

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC QUALITY INSPECTED 2

THESIS

**REENGINEERING THE UNITED STATES MARINE
CORPS' RECRUIT DISTRIBUTION MODEL (RDM)**

by

Kevin J. Snoap

September 1998

Thesis Advisor:
Associate Advisor:

Hemant K. Bhargava
Suresh Sridhar

Approved for public release; distribution is unlimited.

19981009 117

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE : REENGINEERING THE UNITED STATES MARINE CORPS' RECRUIT DISTRIBUTION MODEL (RDM)			5. FUNDING NUMBERS	
6. AUTHOR(S) Kevin J. Snoap				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) The United States Marine Corps accomplishes its mission "to put the right Marine in the right place at the right time with the right skills and quality of life" in a variety of ways. One of the information systems assisting the Marine Enlisted Assignments branch is the Recruit Distribution Model (RDM). This thesis proposes changes to the RDM user interface, data management, assignment model, and analysis capability. With the use of business process reengineering, process modeling, mathematical modeling, and database design a fully functional prototype has been developed to address each identified change proposal. This reengineered system includes numerous innovations such as an intuitive navigational scheme using switchboards, and the elimination of manual data entry for data already available in the system. It also provides a number of significant contributions beneficial to the USMC. For instance, the reengineered system allows the user to objectively analyze different results by comparing four different objective measures, and its mathematical model uses commercial-off-the-shelf products eliminating a proprietary solver. All these changes will empower managers to effectively and efficiently manage the assignment of recruits in order to meet the challenges of the 21st century.				
14. SUBJECT TERMS USMC, Databases, Manpower Assignment, Models, Decision Support Systems, Graphical User Interface			15. NUMBER OF PAGES 100	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

Approved for public release; distribution is unlimited.

REENGINEERING THE UNITED STATES MARINE CORPS' RECRUIT DISTRIBUTION MODEL (RDM)

Kevin J. Snoap
Lieutenant, United States Navy
B.S., University of Texas at Austin, 1991

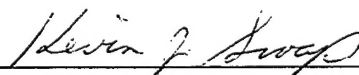
Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
MANAGEMENT

from the

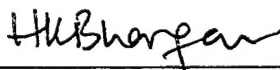
NAVAL POSTGRADUATE SCHOOL
September 1998

Author:

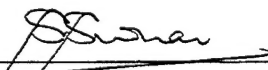


Kevin J. Snoap

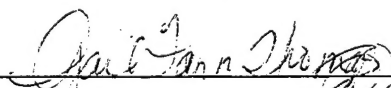
Approved by:



Hemant K. Bhargava, Advisor



Suresh Sridhar, Associate Advisor



Rueben T. Harris, Chairman, Department of Systems
Management

ABSTRACT

The United States Marine Corps accomplishes its mission "to put the right Marine in the right place at the right time with the right skills and quality of life" in a variety of ways. One of the information systems assisting the Marine Enlisted Assignments branch is the Recruit Distribution Model (RDM). This thesis proposes changes to the RDM user interface, data management, assignment model, and analysis capability. With the use of business process reengineering, process modeling, mathematical modeling, and database design a fully functional prototype has been developed to address each identified change proposal. This reengineered system includes numerous innovations such as an intuitive navigational scheme using switchboards, and the elimination of manual data entry for data already available in the system. It also provides a number of significant contributions beneficial to the USMC. For instance, the reengineered system allows the user to objectively analyze different results by comparing four different objective measures, and its mathematical model uses commercial-off-the-shelf products eliminating a proprietary solver. All these changes will empower managers to effectively and efficiently manage the assignment of recruits in order to meet the challenges of the 21st century.

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	THESIS PURPOSE	1
B.	GENERAL PROBLEM DESCRIPTION	1
C.	SIGNIFICANT CONTRIBUTIONS	5
II.	REENGINEERING MOTIVATIONS AND PROCESS VIEW	7
A.	LIMITATIONS OF OLD SYSTEM	7
1.	Navigation	7
2.	Data Management	8
3.	Assignment Procedure	8
4.	Transaction Processing Approach	9
B.	A PROCESS VIEW	9
1.	Recruit Distribution Model Operating Environment . . .	10
2.	New System Level 0 Diagram	11
III.	MODELING THE RECRUIT DISTRIBUTION PROBLEM .	15
A.	OPTIMIZATION OBJECTIVES: FIT AND FILL	15
B.	MEASURING FITNESS	16
C.	MEASURING FILL	18
D.	ASSIGNMENT MODEL	19
1.	Assumptions	19
2.	Notation	20
3.	Objective	20
4.	Constraints	20
IV.	A DECISION SUPPORT SYSTEM FOR RECRUIT DISTRI- BUTION	23
A.	ARCHITECTURE AND IMPLEMENTATION	23
1.	Architecture	23

2.	Implementation	26
B.	USING THE RECRUIT DISTRIBUTION DECISION SUPPORT SYSTEM	28
1.	Setting up a Run	28
2.	Model Execution	31
3.	Customizing a Run and Results	33
C.	INNOVATIONS IN USER INTERACTION	34
D.	INNOVATIONS IN ANALYZING MODEL RESULTS	35
E.	OBJECTIVE COMPARISON OF OLD AND NEW SYSTEM SOLUTIONS	39
F.	SUMMARY	41
V.	CONCLUSION	43
A.	SIGNIFICANT CONTRIBUTIONS	43
B.	LESSONS LEARNED	43
C.	PROTOTYPE IMPROVEMENTS	44
1.	Speed Improvements	44
2.	Multiple Solution Storage	45
3.	Administration Switchboard	46
	APPENDIX A. ACRONYMS	47
	APPENDIX B. AS-IS RD BUSINESS PROCESS (IDEF0) MODEL	49
	APPENDIX C. TO-BE RD BUSINESS PROCESS (IDEF0) MODEL	61
	APPENDIX D. RDDSS VBA CODE	67
	LIST OF REFERENCES	87
	INITIAL DISTRIBUTION LIST	89

LIST OF FIGURES

1.	Assigning Marines to Schools	2
2.	Assigning Marines to School Classes	4
3.	Current Recruit Distribution Operating Environment	10
4.	RDdss Process, Level 0 Diagram	12
5.	Exponential Function for Calculating $Score_{level(p)}$	17
6.	Two Candidate Penalty Functions	19
7.	RDdss Architecture	23
8.	RDdss Main Switchboard	24
9.	RDdss Relational Schema	25
10.	RDdss Demonstration Solver	26
11.	RDdss Decomposition Diagram	27
12.	Actual RDdss Build Sequence	28
13.	Setting up a Model Run	29
14.	Model Execution	32
15.	Changing a Graduation Date	35
16.	Two alternative results: Fit	37
17.	Two alternative results: Fill	37

I. INTRODUCTION

A. THESIS PURPOSE

The main purpose of this thesis was to reengineer a United States Marine Corps' Manpower Assignment model concerned with the distribution of recruits to schools. This model is called the Recruit Distribution Model (RDM) [Ref. 1]. Throughout this thesis, the RDM is addressed as either RDM or the old system.

Additionally, an important purpose of this thesis was to build a functional prototype of the reengineered RDM. The new system is called the Recruit Distribution Decision Support System (RDdss). It demonstrates the functionality of the reengineered RDM. Throughout this thesis, the RDdss is addressed as either RDdss or the new system.

The majority of this thesis is devoted to a discussion of the RDdss. As necessary, the RDM is discussed. The following section is a general discussion designed to set the stage for understanding the rest of the thesis. It provides a problem description of recruit distribution in the United States Marine Corps (USMC). The last section of this chapter discusses the significant contributions of this thesis.

B. GENERAL PROBLEM DESCRIPTION

Recruit distribution in the USMC is the process that assigns recruits to an entry level school (ELS) leading to a military occupational specialty (See Figure 1). These assignments are made about 48 times a year, during the last week of Marine Corps Recruit Depot (MCRD) training. In this ending period of the MCRD, the recruits are facing the "Crucible," which is the final wicket a recruit endures before officially becoming a Marine. Consequently, the use of the titles recruit and Marine are used interchangeably in this paper.

For at least two reasons, this assignment process is a critical manpower function. First, a Marine's military occupational specialty (MOS) ultimately determines

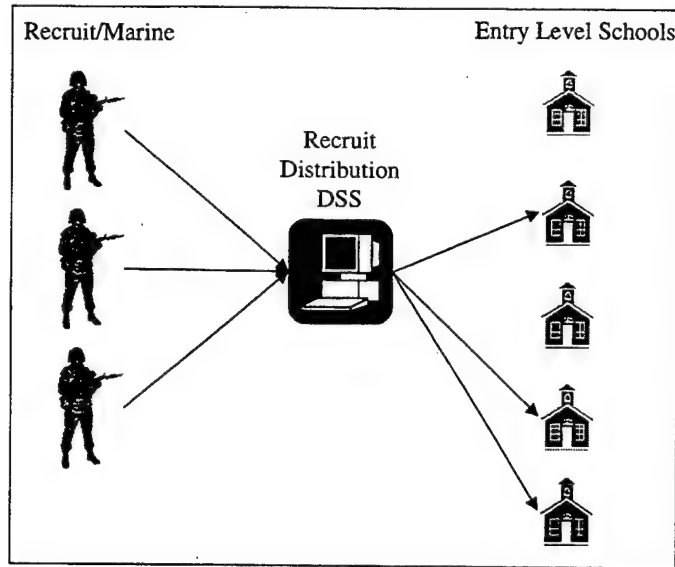


Figure 1. Assigning Marines to Schools

the member's career. Therefore, it is in the best interest of both the Marine and the USMC that a school assignment matching the Marine's desire is made. Second, the success obtained by a service member during his or her time in the USMC is partially based upon successful completion of their ELS, where a pattern of success is established. Therefore, a school assignment maximizing the chances of the Marine completing their training is important. So, fulfilling the Marine's desire and matching him or her to an MOS (i.e. ELS) that fits their personal characteristics is critical to the overall health of the USMC, making the assignment process a critical manpower function [Ref. 2].

Fulfilling the desire of the Marine is accomplished through the use of a contract guarantee. This is called a program enlisted for (PEF), and is specified during the recruiting process. For instance, a PEF = 19 is the "Tank and Assault Amphibian Option." There are currently two ELS's associated with this, "M1A1 Tank Crewman" and "Assault Amphibian Crewman." So, a Marine who chose the PEF = 19 is a possible candidate for these two schools and no others.

Once the Marine's school options are known, a Marine-to-school fit is deter-

mined for each of these schools (See Chapter III, Section B for details). This fitness determination is partly made by looking at each ELS's minimum eligibility requirements, which are called mandatory properties. This term "property" is used as defined by Webster's Ninth New Collegiate Dictionary: "a quality or trait belonging and especially peculiar to an individual or thing." Examples are Age > 18 , Clerical ≥ 80 , and Electrical ≥ 95 . The meaning of the first example is obvious. The other two are based on test scores from the Armed Services Vocational Aptitude Battery (ASVAB) test.

In addition to the mandatory properties, most schools also specify desirable properties. A desired property is the same as a mandatory property, except they are not prerequisites for attending the school. For example, the Traffic Management Coordination school desires Marines with a Clerical score of at least 100. So, a desired property of Clerical ≥ 100 is specified for this ELS.

By using the information obtained from the PEF, mandatory and desired properties, a fitness matrix is generated. This shows the fit of every Marine to every school he or she is eligible for. Since there are about 100 schools and on average 700 Marines considered for every run, this matrix has the potential of 70,000 matches.

However, the matrix size is actually bigger. This is because each of the schools is broken down by classes. Some ELS's have a class starting each week, others every month, and others every quarter. Over a given year, these classes total about 1,800. Following the practice of the USMC, only the classes over the next 3-4 months are considered during the assignment process (see Figure 2). Therefore, the fitness matrix is increased to a potential size of 350,000.

The fact that the school classes start at different times throughout the year and that the assignment process is only conducted 48 times a year (normally this occurs on Friday) causes the "3-month look ahead" in the model and has implications important to the USMC. The concern is school seats which never get filled. Each seat is prepaid, guaranteeing its availability to the USMC. This means every vacant

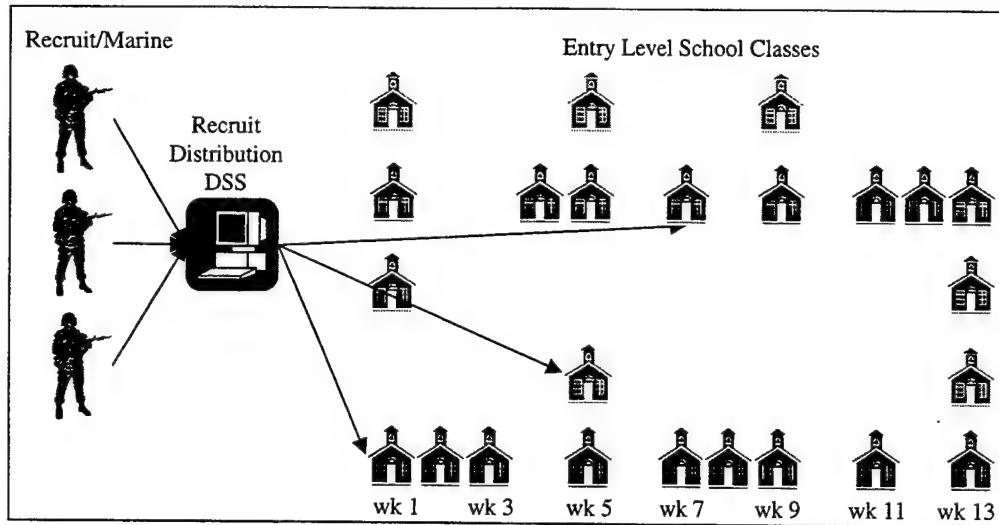


Figure 2. Assigning Marines to School Classes

spot is potentially a lost resource to the USMC. Therefore, in addition to the Marine-to-school fitness concern, there is also the concern of filling all available school seats before their report date is passed.

Another concern is the problem of unassigned Marines. There are numerous reasons why they may not get assigned. Maybe the only school class the Marine was eligible for is already full. Or, possibly he or she is not qualified for any of the school's promised by their PEF. A third possibility is errors in the data provided to the assignment process. Regardless of the cause, it requires identification and corrective action.

Finally, a discussion of the internal and external stakeholders associated with the USMC recruit distribution is given. Internally, there are the USMC and the recruits. Since their concerns were discussed above, they are not given any further consideration. Externally, there is the contractor Decision Support Associates, Inc. (DSAI) who has maintained and upgraded the RDM for over 30 years. They also maintain about eight other major systems for the USMC. For their services, the USMC pays a significant amount of money each year. Additionally, DSAI has proprietary software in some of these systems, which has locked-in the USMC to this

company. Besides the contractor, there is one last significant stakeholder, the American taxpayer. In a time of increasing fiscal constraints and shrinking budgets, it is imperative that wise decisions are made in regards to assigning Marines to ELS's. The American taxpayers expect nothing less.

C. SIGNIFICANT CONTRIBUTIONS

Four significant contributions beneficial to the USMC have been made in this thesis. A detailed discussion of these is given throughout the next three chapters. Here, we list and summarize the contributions.

- Analysis and articulation of the recruit distribution process - by interviewing USMC and DSAI personnel, reviewing available documentation, and operating the RDM we were able to articulate an understanding of the recruit distribution process using IDEF process modeling. Additional data modeling was articulated in a third normal form relational schema [Ref. 3].
- Development of a mathematical model - by analyzing the assignment process, criteria, and constraints, USMC policies and objectives, a mathematical model for the new system was developed.
- Fully functional prototype - an intuitive and easy to understand new system that seamlessly interfaces with the old system was built. It provides an objective means of comparing assignment results of both systems, and was developed using commercial-off-the-shelf (COTS) software applications.
- Elimination of the proprietary solver and associated contractor lock-in - this was accomplished by replacing the proprietary solver with two COTS applications.

II. REENGINEERING MOTIVATIONS AND PROCESS VIEW

The RDM has a number of limitations which motivated the USMC to reengineer it. We start this chapter by listing and describing each identified limitation. Then, to develop a better understanding of recruit distribution in the USMC, two different process views are examined. The first is concerned with the RDM operating environment, and the second is concerned with the first level of the RDdss IDEF0 model.

A. LIMITATIONS OF OLD SYSTEM

Throughout this reengineering effort, a number of limitations of the old system have become apparent. Many of these limits were recognized earlier, and were part of the USMC's motivation to reengineer the RDM. The following is a list of the identified limitations of the old system.

- Navigation
- Data management
- Assignment procedure
- Transaction processing approach

A description of each limitation is now given.

1. Navigation

Navigating through the RDM is not intuitive. It is neither obvious where one should start or where one should go. Navigation is accomplished by initially selecting an option from one of the main display's drop down menus. This normally results in a window appearing on the computer's desktop. Then, by pointing and clicking on the displayed window's buttons, further navigation is accomplished. After working with the RDM for a number of hours, we were able to navigate through the application

to find and display specific windows. However, after a week or two of not using the system, we had difficulty finding our way around again. In the RDdss, we have created an intuitive navigational scheme using switchboards.

2. Data Management

Data management in the RDM is poor, leading to the introduction of numerous errors. The biggest problem in this regard deals with data entry. The RDM violates the basic rule of never requiring the user to enter data already in the system [Ref. 4]. For example, creating a new school in the RDM requires the user to type data into seven different data fields. Only the course identification number has been automated by a drop down list. In the RDdss, when creating a new school we have reduced the manual data entry to one field.

3. Assignment Procedure

There are at least five limitations associated with the old system's assignment procedure. It

- is encoded into proprietary software,
- examines schools sequentially, rather than globally,
- makes assignments based on school priority rather than weights,
- attempts to maximize fill rather than fit-and-fill, and
- is relatively inflexible.

The solver performing these assignments was designed back in the 1950's, where the major concern was speed and using the minimal amount of memory. It is written in Fortran, and is proprietary code owned by DSAI. It does not search repetitively for an optimal solution by trying to maximize or minimize an objective function. Instead, it maximizes the fill of prioritized school seats. In the RDdss, we use a well know algorithm called CPLEX [Ref. 5]. After conducting around 600 iterations of comparing 344 Marines to 576 school classes, it produces an *optimal* solution. Its

objective function was written to make the assignment procedure flexible. It allows the RDdss manager to “game” the system by making fit-and-fill trade-offs, until a “good” solution is found.

4. Transaction Processing Approach

Finally, the old system follows a transaction processing approach, vice a decision support approach [Ref. 6]. This is partially due to the inflexibility of the solver. However, another contributor is the extreme difficulty in comparing one run to another. Other than providing a means for manually computing the numerical difference in Marines assigned, the RDM provides little insight or information views for comparing runs. This is because there is no way to objectively compare one run to another. As it is now, if the RDM manager launches the assignment model and everybody is assigned, the result becomes the approved assignments. In the RDdss, we have created an entire process devoted to providing insightful analysis of a given run (See Chapter IV, Section D for details). Its purpose is to support the making of wise assignment decisions.

B. A PROCESS VIEW

To develop a better understanding of recruit distribution in the USMC, two different process views are now examined. The first is concerned with the current operating environment of the old system. The goal is to develop a big picture view of this process. For those interested in further study of the old system, Appendix A contains the RDM business process IDEF0 model.

The second process view examined concerns the first level of the RDdss business process IDEF0 model. The goal is to develop a better understanding of the new system, without going into great detail. This lays a good foundation for the RDdss discussions in the remainder of this thesis. Appendix B contains the entire IDEF0 model of the new system.

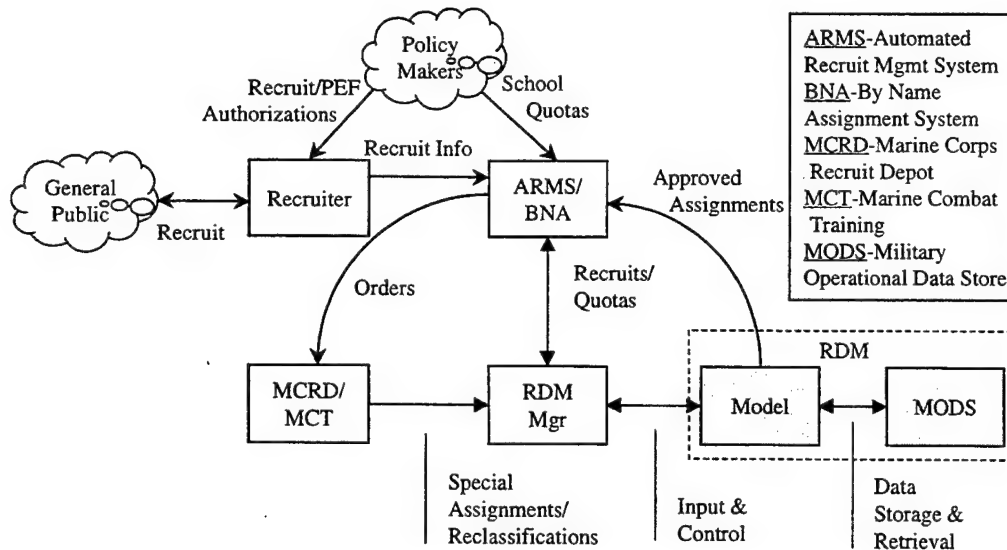


Figure 3. Current Recruit Distribution Operating Environment

1. Recruit Distribution Model Operating Environment

Figure 3 graphically depicts the operating environment of the USMC recruit distribution. At the top are the policy makers, who forecast and decide how many recruits and corresponding school seats are needed over the next few years. In addition to this, they determine how many PEF's or school guarantees to make available for a given year. Currently, about 65-70% of the recruits enter the USMC with a guarantee. The recruit and PEF authorizations for the following year are given to the USMC recruiters, and the school seats or quotas determined for the following years is input into the By Name Assignment (BNA) system.

The recruiters use the recruit and PEF authorizations in recruiting from the general public. Once a potential candidate is found they take the ASVAB, if they have not already done so. Based on the results, the recruiter can offer different guarantees. Once the candidate signs a contract with the USMC, their test scores, PEF, and other personal information such as age and height are entered into the Automated Recruit Management System (ARMS).

Both the ARMS and BNA systems utilize large main frame databases. They

serve as central repositories for maintaining data on many aspects dealing with the USMC. Keeping this information stored in one location ensures everyone is using the same data, providing consistency for all users. One user is the RDM manager, who retrieves recruit and school information from these systems for use in the RDM.

Additionally, the RDM receives data from two other sources. The MCRD instructors provide special assignment inputs. These are personnel identified as having the talents or abilities well suited for a particular school. The other data comes from the MCT. They provide the RDM manager with reclassification information. For instance, a Marine is reclassified if he or she is injured during MCT and is unable to make the start date of their assigned ELS.

All this recruit and school data is input into the RDM, where it is stored in the Military Operational Data Store (MODS). The model is then run. Once a satisfactory set of assignments is obtained, the RDM manager uploads the approved assignments to the ARMS and BNA systems. From this assignment information, the MCRD generates orders for the Marines graduating from the MCRD.

2. New System Level 0 Diagram

Focus is now shifted to the examination of the first process level of the RDdss. Looking any deeper will provide more detail than is necessary at this point. As mentioned earlier, the intent is to lay the foundation for the RDdss discussions in the remainder of this thesis.

Figure 4 shows the level 0, or context diagram of the RDdss. The program used to generate this illustration was BPWin, which is based on IDEF0 modeling. Consequently, all arrows entering from the left are considered "inputs," arrows entering from the top are "controls," arrows exiting to the right are "outputs," and arrows entering from the bottom are "mechanisms." A convenient acronym for this is ICOM [Ref. 7].

- I = Input: something consumed in the process
- C = Control: a constraint on the operation of the process

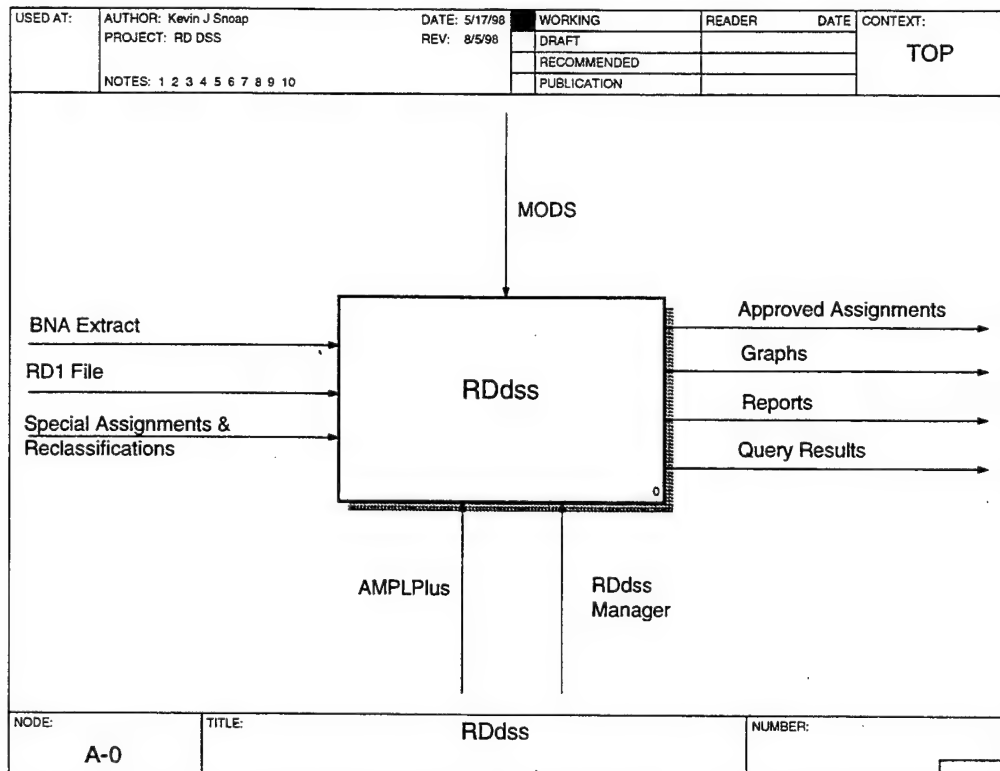


Figure 4. RDdss Process, Level 0 Diagram

- O = Output: something resulting from the process
- M = Mechanism: something used to perform the process, but is not itself consumed

There are three inputs to the RDdss, the BNA extract, the RD1 file, and the special assignments/reclassifications. The first of these contains a set of school class quotas covering a 120 day period. This comes to the RDdss manager as a fixed-width delimited text file. The RD1 file contains the data for the recruits soon to graduate from the MCRD. It also is a fixed-width delimited text file. The special assignments and reclassifications are the recruits identified with special abilities or talents and the Marines requiring reassignment to a different school class, respectively. This last data comes to the RDdss manager as a memo and not a text file.

The only control is the database (or Military Operational Data Store) used to store the data for the RDdss. Among other things, the MODS contains the informa-

tion specifying the eligibility requirements for each of the schools. This is why it is considered a control. It contains the data controlling or constraining the assignment of Marines to schools.

Besides the approved assignments that were mentioned earlier, the output consists of graphs, reports, and query results. Each graph was designed to provide insight into the current assignment result, plus provide a means for comparing different runs. The reports provide a print-out of information the RDdss manager might find useful. For instance, an approved assignment report and an unassigned Marine report are both available. The query results provide different views of the data in the MODS. For instance, one of the queries provides a listing of the fitness scores of all schools a Marine is eligible for.

Finally, the two mechanisms of the RDdss are the RDdss manager and the commercial-off-the-shelf (COTS) application AMPLPlus. Both of these entities are necessary to make the system work, and are not consumed by the process itself. This is why they are considered mechanisms. The RDdss manager is the one who points and clicks on the buttons of the RDdss, making it operate. AMPLPlus is the application containing the mathematical model used in making optimal Marine-to-school assignments. It interfaces with the CPLEX algorithm that actually performs the optimization.

III. MODELING THE RECRUIT DISTRIBUTION PROBLEM

We model the recruit distribution problem as an assignment problem. Various submodels are developed to compute input parameters and perform other preprocessing steps for the assignment model, which optimally assigns recruits to a school class. Following is a discussion of important points concerning the recruit distribution problem.

1. Schools and Classes: there are about 130 schools per fiscal year, which are broken down by classes (about 1800 classes per fiscal year). Classes of the same school, commencing on different dates, are identical. However, a class commencing soon after a recruit is available for training has greater utility than another class starting at a later date. On average, there are 600 classes in a run.
2. Program Enlisted For (PEF)
 - (a) 65-70% of enlistees enter the USMC under a guarantee (or PEF). The remaining recruits enter under an open contract (PEF=00).
 - (b) A recruit can be assigned only to schools associated with his or her PEF. Some PEF's are associated with one school, while others are associated with many schools.

A. OPTIMIZATION OBJECTIVES: FIT AND FILL

Our approach in the assignment model is to optimize by looking at both fit and fill. The importance of both these objectives was discussed in Chapter II, Section

B. The idea is to allow the RDdss manager to make a tradeoff between fit and fill.

The high-level objective function is:

- Maximize

$$K_1 \cdot \text{Fitness} - K_2 \cdot \text{Penalty}$$

Where K_1 is the fitness coefficient and K_2 is the fill coefficient.

By assigning different values to these coefficients, the RDdss manager is able to make trade-offs between fit and fill. This capability makes the assignment model flexible. It provides the RDdss manager a means for “gaming” the system.

B. MEASURING FITNESS

The first half of the high-level objective function deals with maximizing the fitness of Marines to schools. Therefore, we cover this topic now. Following is a discussion of important points concerning the measuring of fitness.

1. Marines are only eligible for assignment to schools corresponding to their PEF guarantees.
2. Associated with each school are specific mandatory properties, or minimum eligibility requirements (e.g. AGE > 18).
3. Eligibility of a Marine to a school is pre-determined by comparing a Marine's attributes (e.g. Age = 20) to the school's mandatory properties. A Marine is eligible only if he or she meets all the minimum school requirements.
4. Many schools also have desirable properties associated with them. This is a property which is desired in a Marine (e.g. Height > 65 inches), but are not prerequisites for attending the school. Since some desirable properties are more desirable than others, a means of distinguishing the properties is necessary. We have followed the Marine Corps practice of using 6 levels of desirability. Level 1 properties are the most desired, followed by level 2, etc.
5. Let $\mathcal{P}(s)$ denote the set of desirable properties for school s , and let $level(p)$ denote the desirability level of property p . Let $possesses_{(m,p)} = 1$ if Marine m possesses property p (0 otherwise). Further, let $Score_{level(p)} (> 0)$ be the score assigned for possessing a $level(p)$ property, where $Score_{level(p)}$ is calculated using an exponential function inversely proportional to $level(p)$. The scores for levels 1 through 6 are shown in Figure 5.
6. The total fitness score ($fit_{m,s}$) of Marine m to school s is composed of two parts: the score ($ManScore_{m,s}$) for possessing mandatory properties, and the score ($DesScore_{m,s}$) for possessing some or all of the desirable properties.

Our procedure for computing fitness scores is designed in a manner that, for each school, the average fitness score—with the average computed over all qualified Marines, i.e. Marines who have met the mandatory properties—is constant (i.e. 100). The reason we did this was to ensure each school

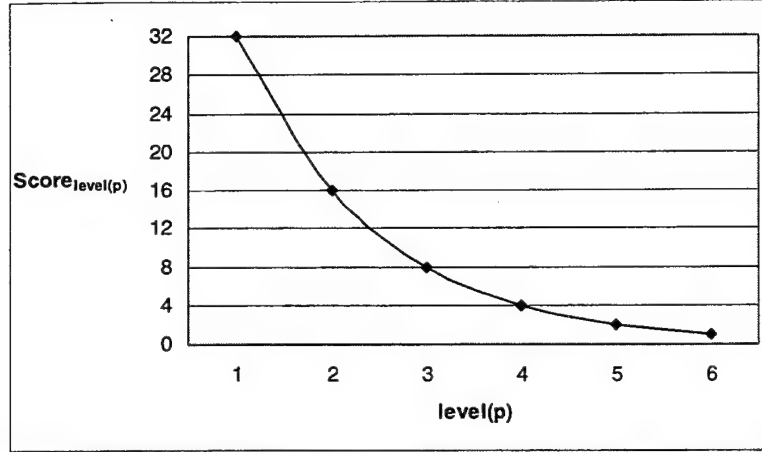


Figure 5. Exponential Function for Calculating $Score_{level(p)}$

received equal treatment in the model, regardless of the number of desirable properties specified for each school. Otherwise, all other things being equal, the school with the greatest number of desirable properties would receive the most assignments.

7. For each school s , percentage weights $ManWt_s$ and $DesWt_s$ are assigned, respectively, for the mandatory and desirable properties. For example, for school s_1 , a weight of 70% may be assigned to the mandatory properties and 30% for the desirable properties.
8. A Marine possessing all mandatory properties for school s is given an initial score $ManScore_{m,s} = ManWt_s \cdot 100$. Marines who do not possess all mandatory properties for a school are given an overall fitness score of 0 for that school.
9. Of all the Marines possessing the mandatory properties, those possessing desirable properties are awarded additional points, which are weighted by the level of the property. The Marine's score for desirable properties is computed as follows:
 - The absolute score $Abs_{m,s}$, for Marine m and school s , based on desirable properties is

$$Abs_{m,s} = \sum_{p \in \mathcal{P}(s)} Score_{level(p)} \cdot possesses(m, p)$$

The average of these absolute scores is computed over all qualified Marines. Let $AveAbs_s$ denote the average absolute score for school s .

- Each Marine's fitness score for desirable properties is then computed as a fraction of this average, and normalized by multiplying with the percentage weight for desirable properties. That is, $DesScore_{m,s}$ is a weighted relative score:

$$DesScore_{m,s} = \left(\frac{Abs_{m,s}}{AveAbs_s} \right) \cdot DesWt_s \cdot 100$$

The overall average of all these scores is, due to the above construction, $DesWt_s \cdot 100$.

10. The final fitness score, $fit_{m,s}$ is simply the sum $ManScore_{m,s} + DesScore_{m,s}$. It may be seen that, for each school, this number averages (over all the qualified Marines for that school) to 100.
11. As mentioned earlier, schools are broken down by classes that are identical, except for their start dates. Therefore, let $fit_{m,s} = fit_{m,c}$ for each class c of school s .

C. MEASURING FILL

The second half of the objective function deals with maximizing the fill of school seats. We now turn our attention to this topic.

As we discussed in Chapter II, school seats are paid for in advance. This is to guarantee the seat is available to the USMC. Consequently, every vacant spot is potentially a lost resource. Since the model is run about once every week, the biggest concern is the unfilled seats having report dates within the next seven days. Conversely, a seat having a report date in three months is not so critical.

To capture the essence of filling school seats with early report dates, we use a penalty function. The idea is simple. School seats having an early report date get a high penalty, and those having a late report date get a low penalty. A number of functions would have worked for this. We have graphed two candidates in Figure 6. The first is a linear function and the second a large-step function. Each of these was tried in the prototype. In our opinion the latter one is the best choice. It gives a lot of emphasis to the first week, plus it treats all schools having report dates within the same week equally. It is the function currently implemented in the rddss.

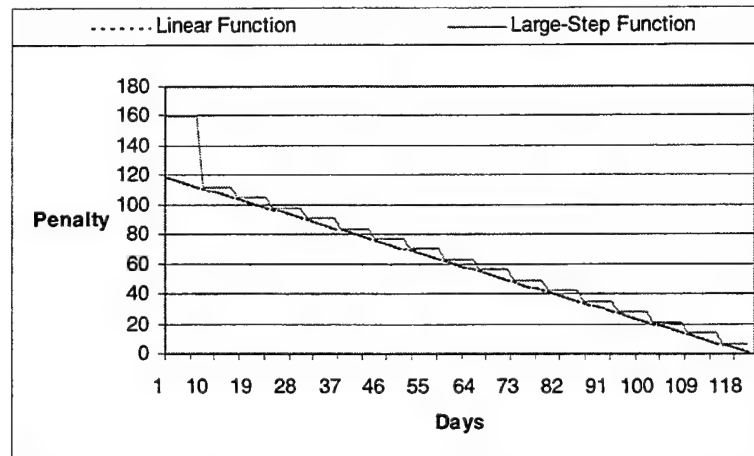


Figure 6. Two Candidate Penalty Functions

D. ASSIGNMENT MODEL

We are now ready to formally describe the assignment model. We will first specify the model's assumptions. Then, we will list the model's notation, followed by its objective function and constraints.

1. Assumptions

1. Each school is unique and identifies a single course of instruction leading to an MOS.
2. In any given run, a school may offer several classes that start on different dates. It is more desirable to fill seats in classes starting earlier. Such classes have a higher penalty per unit shortfall than similar classes starting at a later date.
3. All Marines available for assignment are graduating from the MCRD on the same date.
4. A Marine's eligibility for a school, as well as the fit for each school, is determined by the model preprocessor, taking into account the PEF guarantees, and the mandatory and desirable properties.
5. A Marine is assigned to a school corresponding to their PEF code, or not at all.
6. It is better to leave a Marine unassigned than to overfill a school.

7. The demand for the number of Marines to be trained in each school class is determined by looking at the BNA extract. This demand statement is already constrained by the capacity constraint at each school.

2. Notation

- Sets

\mathcal{M} : Marines

\mathcal{C} : Classes

- Exogenous Variables (Parameters in AMPL)

$fit_{m,c}$: the desirability of assigning Marine m to class c (note: the fitness score is zero for Marine-class pairs where either the Marine does not meet the classes mandatory properties, or where the class does not fulfill the Marine's PEF guarantee.)

$demand_c$: demand for Marines to be trained at class c

$penalty_c$: penalty for each unit of demand not met (higher value means it is more critical to fill the class)

- Decision Variables

$x_{m,c}$ (binary integer): 1, if Marine is assigned to the given class; 0 otherwise

3. Objective

Maximize the Total Utility: (Total Reward - Total Penalty)

$$TU = k_1 \left(\sum_{c \in \mathcal{C}} \sum_{m \in \mathcal{M}} fit_{m,c} \cdot x_{m,c} \right) - k_2 \left(\sum_{c \in \mathcal{C}} penalty_c \cdot (demand_c - \sum_{m \in \mathcal{M}} x_{m,c}) \right) \quad (\text{III.1})$$

4. Constraints

- assignmentLimit: a Marine is assignable to at most one school

$$\sum_{c \in \mathcal{C}} x_{m,c} \leq 1 \quad \forall m \quad (\text{III.2})$$

- eligibility: a Marine is only assignable to a class they are fit for (this prevents assigning Marines with a fitness score of zero)

$$x_{m,c} \leq fit_{m,c} \quad \forall m \quad \forall c \quad (\text{III.3})$$

- capacity: since penalties apply only if there is a positive shortfall, the objective function would be non-linear. To avoid that, we assume we will never oversupply Marines to schools, resulting in:

$$\sum_{m \in \mathcal{M}} x_{m,c} \leq demand_c \quad \forall c \quad (\text{III.4})$$

IV. A DECISION SUPPORT SYSTEM FOR RECRUIT DISTRIBUTION

A. ARCHITECTURE AND IMPLEMENTATION

Following is a discussion of the RDdss architecture and its implementation. First, the architectural components are discussed. Then, the steps taken to build the RDdss are described.

1. Architecture

The architecture of the RDdss is depicted in Figure 7. This illustration shows how the six major components of the system are related. We will describe this architecture by examining each component in the following order: switchboard, relational database, preprocessor, assignment model, solver, and analyzer.

The switchboard is one of two components providing an interface between the user and the RDdss. It is the mechanism by which the user controls the operation of the system. Figure 8 illustrates the main switchboard of the RDdss. It is displayed

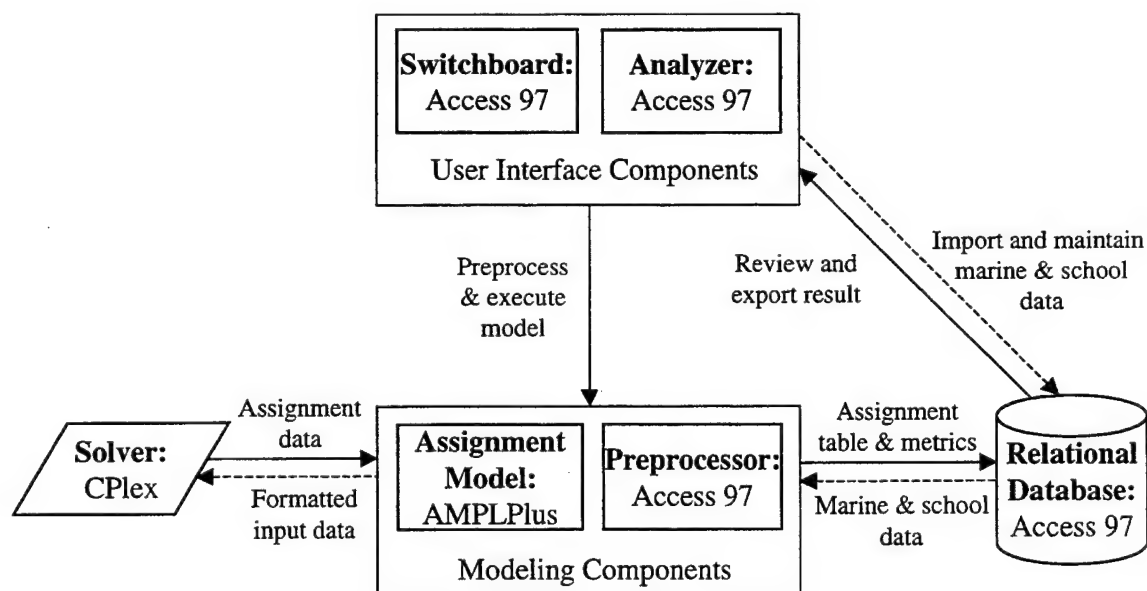


Figure 7. RDdss Architecture

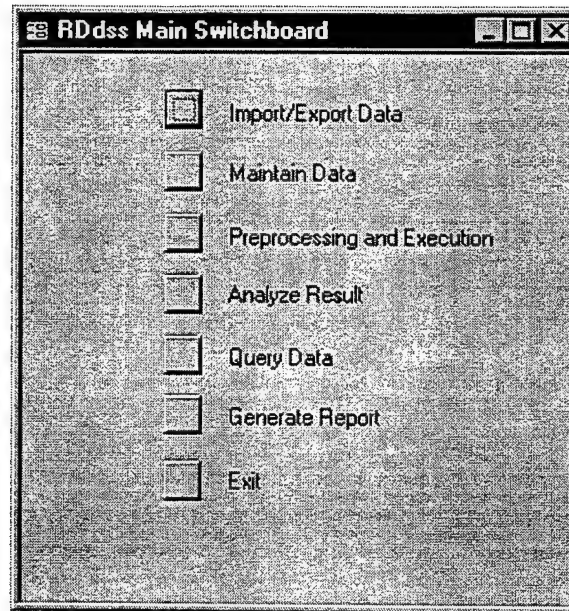


Figure 8: RDdss Main Switchboard

when the application is started. As this illustration depicts, the user can perform the following functions by simply pointing and clicking on the appropriate button: import and export data, maintain data, preprocessing and execution of the assignment model, analyze results, query data, and generate reports. All the RDdss switchboards were generated using Access 97.

The relational database technology was used in the RDdss [Ref. 8]. The actual table relationships are shown in Figure 9. These tables were developed in Access 97, which uses the relational technology. In addition to a few flat files used to communicate with the assignment model, all the data for the RDdss is stored in this database.

The preprocessor is one of the two modeling components. It consists of Access 97 Visual Basic for Applications (VBA) code that computes information necessary for the assignment model [Ref. 9]. It determines the demand of each class, the appropriate penalty associated with each class, and the Marine-to-class fitness matrix. The code for these assignment model inputs is found in Appendix D, lines 452-516 and 4305-4812.

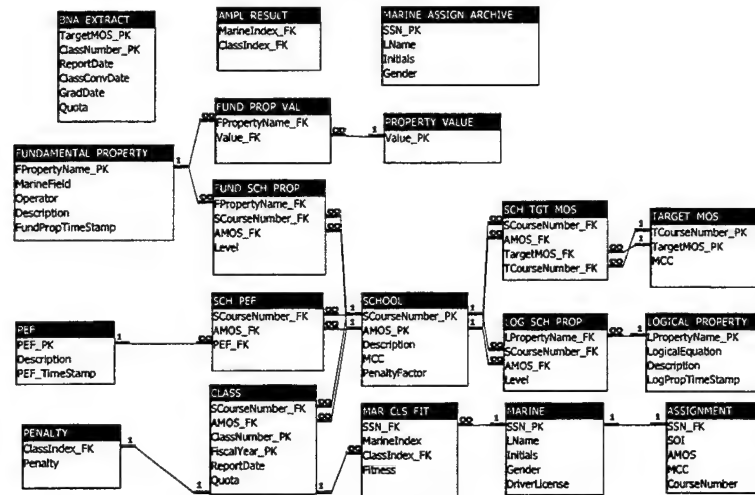


Figure 9. RDdss Relational Schema

The assignment model is the other component of the modeling components. It has been modeled in AMPL [Ref. 10], a language and a computing environment for expressing, solving, and analyzing mathematical programming problems (minimizing or maximizing a function of decision variables, subject to constraints on the variables). A complete discussion of the assignment model's math model is available in Chapter III, Section D.

The next component is the solver. It communicates directly with AMPL, the modeling language of AMPLPlus. As mentioned in Chapter II, it uses an algorithm called CPLEX, which is an optimization package for linear, network and integer programming. Working in conjunction with AMPLPlus, the solver finds an optimal solution.

The final component is the analyzer. It is the other component making up the user interface. This is where the RDdss manager analyzes assignment results, seeking insights for developing a "good" solution. Insight is developed mostly by the graphs and numerical information available in the analyzer. These were generated in Access 97 using its report generator and VBA code. The programming used in the analyzer is shown in Appendix D and includes lines 1-310, 2156-2214, and 3756-3827.

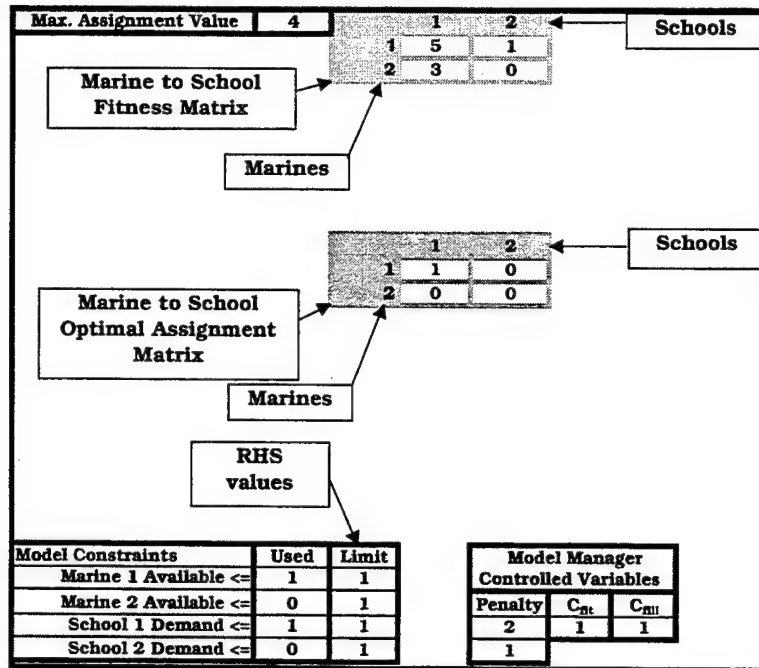


Figure 10. RDdss Demonstration Solver

2. Implementation

Building the RDdss was an iterative process. Some steps took numerous iterations, and some were straightforward. Below is a listing of the 14 major steps it took to build the RDdss.

1. Analyzing the current system was the first step. This included 1) interviewing USMC and DSAI personnel, 2) reviewing available documentation, and 3) operating the RDM.
2. After gathering the current system data, it was necessary to develop a logical understanding of the system. This was accomplished by modeling it with BPWin. Appendix B is the result of this step.
3. With this system understanding, it was possible to envision how the new system would make assignments. Using standard optimization techniques [Ref. 11], a small-scale demonstration solver was built using Excel 97. It is shown in Figure 10.
4. Combining the system understanding and the data requirements for the demonstration solver, it was possible to envision a data structure for the reengineered system. This led to the development of a third normal form relational schema for the RDdss [Ref. 3].

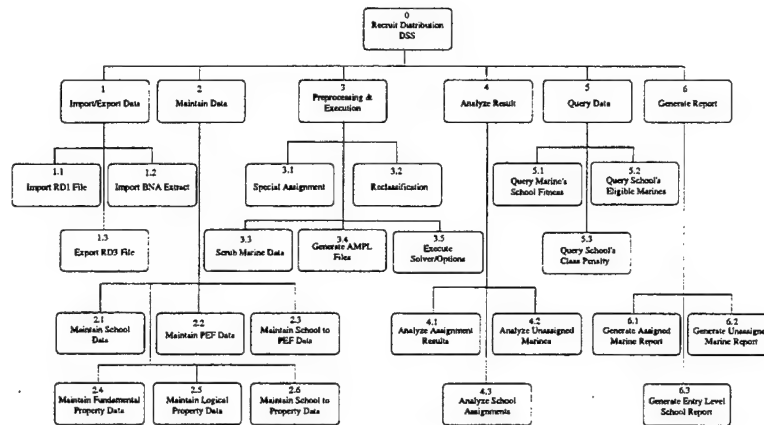


Figure 11. RDDss Decomposition Diagram

5. Using the relational schema as a model, the RDDss tables were built in Access 97. The results of this were seen earlier in Figure 9.
6. As the vision for the reengineered system came more into focus, an intuitive navigational scheme was conceived. This led to the development of a decomposition diagram for the RDDss. It is demonstrated in Figure 11.
7. Using the decomposition diagram as a model, the user interface was built for the RDDss in Access 97. The Main Switchboard was seen earlier in Figure 8.
8. With the switchboards now in place, VBA coding was commenced [Ref. 9]. This made each switchboard functional and useful.
9. Building upon the concept of the demonstration solver from step 3, a formal mathematical assignment model for the RDDss was developed. This was introduced in Chapter III, Section D.
10. The COTS applications, AMPLPlus and CPLEX were installed on the personal computer containing the RDDss.
11. With AMPLPlus now installed, it was possible to code the assignment model into the application. This made it possible to verify our design.
12. Manually creating input files for AMPLPlus is time consuming. So, the preprocessor was coded. This automated the generation of the class demand, class penalty, and Marine-to-class fitness matrix.
13. Further automation of the RDDss was accomplished by interfacing Access 97 and AMPLPlus. This required both VBA and AMPL coding. The end result was a seamless operation of the two applications.

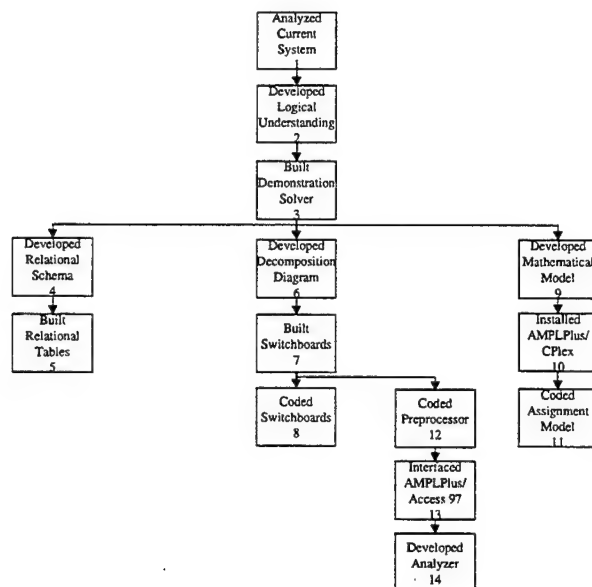


Figure 12. Actual RDdss Build Sequence

14. Finally, much thought was given to ways of gaining insight into the assignment results. This lead to the development of the analyzer, which was discussed briefly in the previous section.

Some of these steps were performed in parallel. The actual order followed in building the RDdss are depicted in Figure 12.

B. USING THE RECRUIT DISTRIBUTION DECISION SUPPORT SYSTEM

In this section, we will discuss using the RDdss. Three aspects are covered, 1) setting up a run, 2) model execution, and 3) customizing a run and results. Each is discussed in the order given here.

1. Setting up a Run

Setting up a run of the RDdss is straightforward and easy to do. Figure 13 graphically illustrates the steps involved. Starting with step 1 on the graph and working through to step 3d(iii), we will discuss a run set-up.

The RDdss manager starts at the top of the Main Switchboard. Here, he or

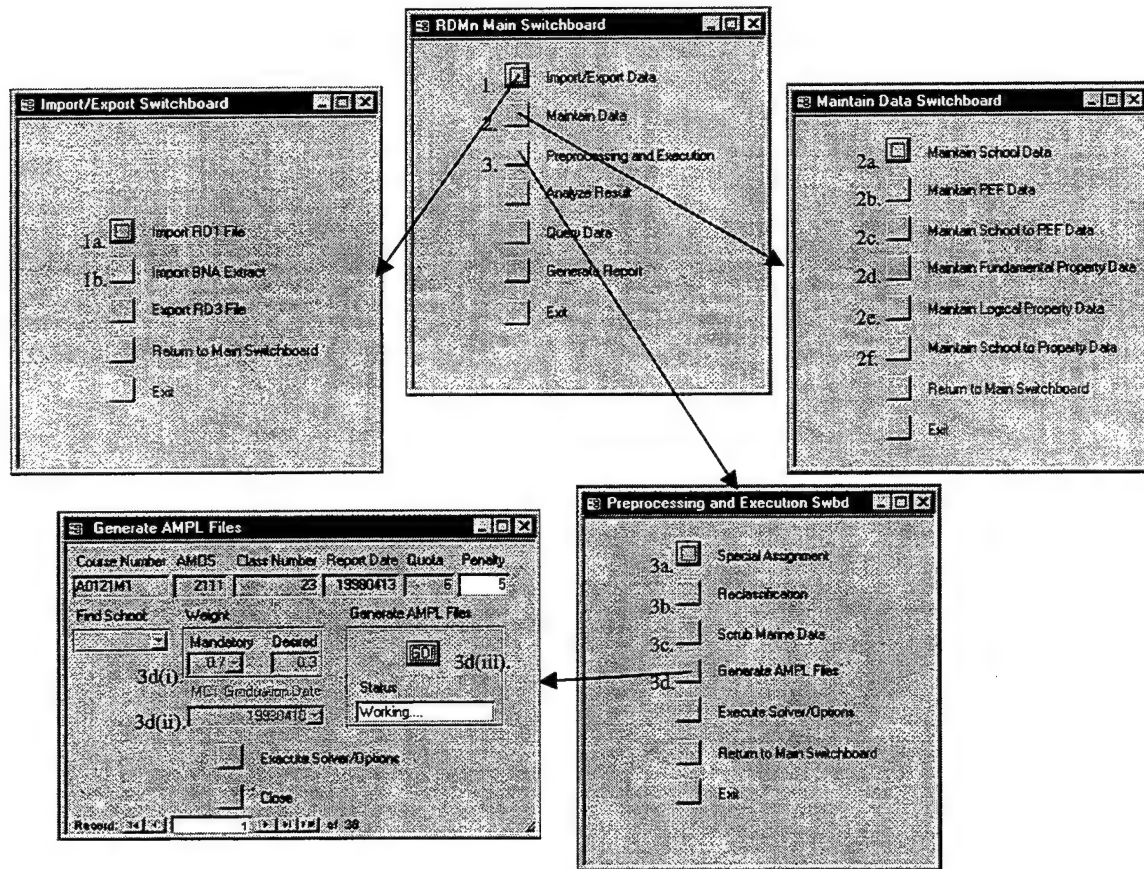


Figure 13. Setting up a Model Run

she clicks on the Import/Export Data button. This will cause the Main Switchboard to close and the Import/Export Switchboard to open. From this switchboard, the RDdss manager will import the RD1 file and the BNA extract into the RDdss MODS. This is easily accomplished by clicking on the corresponding buttons for importing these files. Finally, the RDdss manager will click on the “Return to Main Switchboard” button, causing the Import/Export Switchboard to close and the Main Switchboard to reopen.

The next step is data maintenance. This includes updating, editing, and/or deleting of school related information. The RDdss manager gets to the Maintain Data Switchboard by clicking on the appropriate button on the Main Switchboard.

From the Maintain Data Switchboard, the RDdss manager will start at the

top and work his or her way down to the bottom. The various buttons on this switchboard open forms where data maintenance is performed. For instance, the first button opens the Maintain School Data form, allowing the RDdss manager to create, edit, and/or delete a school. The other buttons allow similar operations with the PEFs, and fundamental and logical properties. After the RDdss manager has finished maintaining the data, he or she returns to the Main Switchboard by clicking on the "Return to Main Switchboard" button.

After data maintenance, the RDdss manager is ready to perform data preprocessing in preparation for a model run. By clicking on the Preprocessing and Execution button, the Main Switchboard closes and the Preprocessing and Execution Switchboard opens. From here, as on all other switchboards, the RDdss manager starts at the top and works down.

The first three preprocessing buttons directly effect the generation of the class demand, class penalty, and Marine-to-class fitness matrix. This is why they come before the fourth button, Generate AMPL Files. Preprocessing in the correct order will reduce the amount of time spent setting up the run, by minimizing the need to generate a new class demand, class penalty, and Marine-to-class fitness matrix.

The first preprocessing button concerns special assignments. Specially assigning a recruit affects the AMPL input in two ways. First, the respective member is not included in the Marine-to-class fitness matrix, since they already have an assignment. Second, the class demand for the class the recruit is assigned to must decrease by one. Otherwise, it may become overfilled.

The second preprocessing button concerns reclassification of a Marine. Reclassified Marines affect the AMPL input in two ways. First, the class demand of the class they are no longer assigned to must increase by one. This provides an opportunity to fill the empty seat during a subsequent run. Second, the class demand of the class the Marine has been reassigned to must decrease by one. Otherwise, it may become overfilled.

The third preprocessing button concerns data scrubbing, which also affects the AMPL input. It is an attempt to identify and correct errors relating to the Marines. An example is a Marine who has been given an unknown PEF. Since it is not identified by the RDdss, the member will receive a zero fitness score for every class. This will prevent the Marine from getting an assignment.

Following the completion of special assignments, reclassifications and data scrubbing, the RDdss manager is ready to generate the AMPL input files. After going to the Generate AMPL Files form, the first concern is specifying the weights for the mandatory and desired properties. This is specified by the RDdss manager at step 3d(i). What this does is indicate how important the mandatory properties are with respect to the desired properties. For instance, if the RDdss manager decided to give a weight of 1.0 to the Mandatory selection, the Desired selection would automatically fill in with 0.0. This would indicate the desired properties are of no consequence. The default for these two selections is 0.7 for Mandatory, and 0.3 for Desired.

Next, at step 3d(ii), the RDdss manager selects the MCT graduation date desired. This is selected from the drop down list. Then, the RDdss manager pushes the "GO!" button. The system will now generate the AMPL input files, completing the run set-up.

2. Model Execution

Execution of the assignment model takes place after the run set-up has been completed. Figure 14 graphically illustrates the steps taken to execute the model. A discussion of this is now given.

From the Main Switchboard, the RDdss manager points and clicks on the Preprocessing and Execution button. This takes him or her to the corresponding switchboard. From here, the Execute Solver/Options selection is made. This opens up the form where the model is executed from.

The Execute Solver/Options form provides the RDdss manager the ability to set the fit and fill coefficients for the objective function. The default values are 1 for

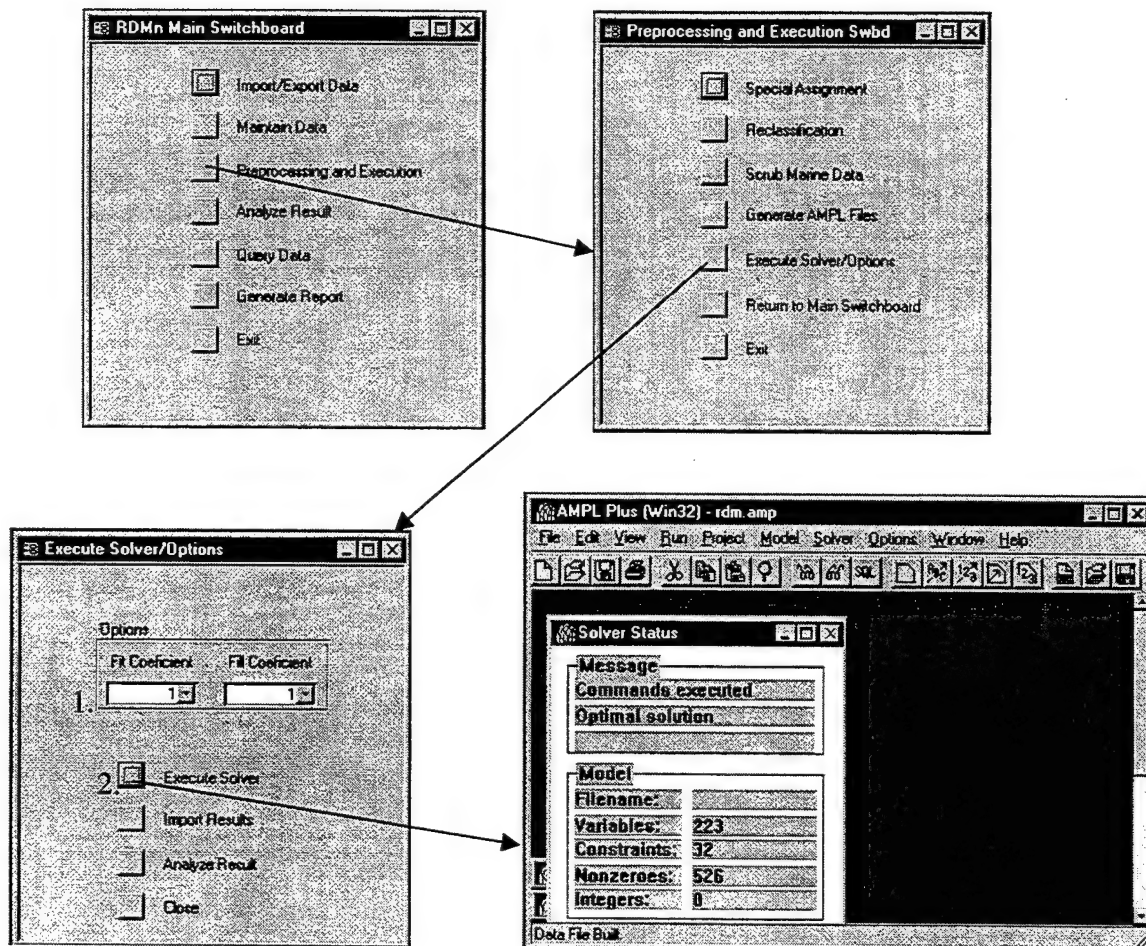


Figure 14. Model Execution

each coefficient. The RDdss manager can change either of these with a whole number from 0 - 100. The significance of these numbers is discussed in detail later in this chapter. Once the desired coefficients have been chosen, the Execute Solver button is pushed.

By selecting the Execute Solver button, AMPLPlus is launched. This application will run without user interaction. First, it will import the RDdss mathematical assignment model. Next, the specified fit and fill coefficients are retrieved. Finally, the preprocessed data generated during the run set-up are imported. Using this input data, AMPLPlus works in conjunction with the solver and generates an optimal assignment solution.

With the generation of an optimal assignment solution, the results are sent to an output file called "rdm.out." AMPLPlus, having completed its task, now closes. This indicates an optimal solution was found and is available for importing. So, the RDdss manager now imports the results by pushing the Import Results button on the Execute Solver/Option form.

3. Customizing a Run and Results

As mentioned in the previous subsection, the RDdss manager can set the fit and fill coefficients of the objective function. This is a key capability of the new system. It is the means by which the RDdss manager can customize a run and "game" the RDdss. This "gaming" of the new system is essential for determining a "good" assignment result.

For instance, if the fit coefficient is set to 0, and the fill coefficient is set to a number > 0 , say 1, the assignment model will maximize the fill of schools having early report dates. This has both advantages and disadvantages. One advantage is the average wait time for Marines to attend a school is minimized. Another is that the number of unfilled school seats, having a report date between the current and next scheduled model run, is minimized. However, a disadvantage is the average Marine-to-school fitness score is guaranteed to be the lowest score of all runs.

Conversely, if the fit coefficient is set to a number > 0 , say 1, and the fill coefficient is set to 0, the assignment model will maximize the average Marine-to-school fitness. An advantage here is the increased chance of each Marine successfully completing their ELS, establishing a pattern of success early in their career. However, one disadvantage is the wait time for a Marine to attend a school is guaranteed to be the longest. Also, the number of unfilled school seats, having a report date between the current and next scheduled model run, is guaranteed to be the greatest.

Neither of the above solutions is ideal. A better solution is some where between these two extremes. By changing the fit and fill coefficients of the objective function, the RDdss manager has a means for customizing each run to ensure a "good"

assignment result is found.

C. INNOVATIONS IN USER INTERACTION

A number of user interaction innovations have been made in the RDdss. Two of them have already been discussed. These were the intuitive navigational scheme using the switchboards, and the elimination of manually entering data already available in the system. One more note worthy innovation is now introduced. It is a date change innovation.

The date change innovation concerns changing the graduation date of the Marines in the downloaded RD1 file. This is necessary for generating correct assignment results. The date in the downloaded RD1 file is based on graduation from the MCRD. However, the model needs to know when the Marines are available for assignment. Availability is based on the MCT graduation date.

The old system solution to this problem took three steps. First, the RDM manager used a calendar to determine the MCT graduation date. This is normally 27 days following completion of MCRD, and is always on a Friday. So, the RDM manager will find the MCRD graduation date on the calendar and count 27 days. The Friday nearest to this day is the day desired. Second, this MCT graduation date, along with the RD1 file, is sent to another branch in the USMC where the date change is made. Third, the file is returned to the RDM manager, who imports the file into the RDM.

The reengineered system has streamlined this process. Within the RDdss, the RDdss manager makes the date change. It is accomplished with the push of a couple buttons.

Figure 15 shows how these changes are made. First, the RDdss manager selects the MCRD graduation date requiring a change. Once this is selected, a calendar appears. Its purpose is to help the RDdss manager correctly identify the MCT graduation date. Next, the "=" button is pushed. Its purpose is to add the number of

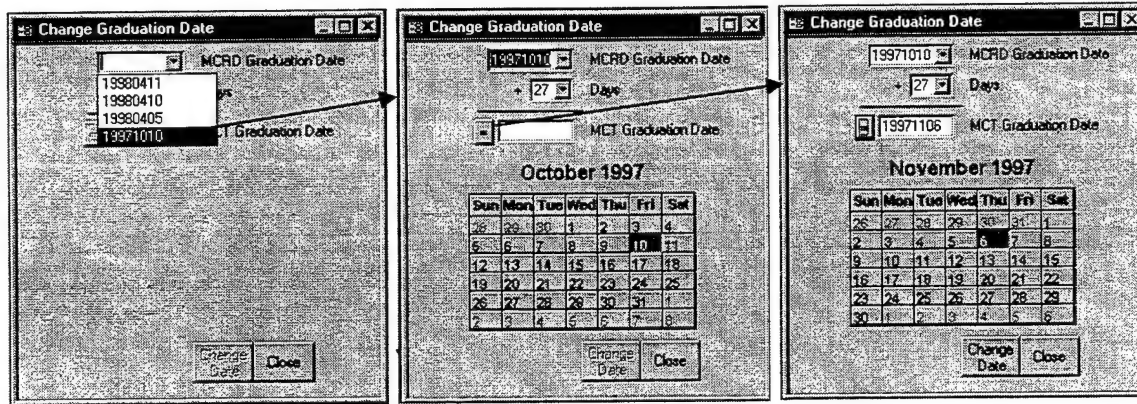


Figure 15. Changing a Graduation Date

days specified in the “Days” field (default is 27), to the MCRD graduation date. This updates the calendar as shown on the far right. When the correct date is found, the “Change Date” button is pushed. This makes the desired change to the data in the RDdss database.

D. INNOVATIONS IN ANALYZING MODEL RESULTS

As discussed earlier, the RDdss manager has the capability of “gaming” the system by using different fit and fill coefficients. However, having this ability is not enough. The RDdss manager still needs a means to compare one run against another.

It is hard, if not impossible, to estimate the quality of a solution simply by looking at a list of assignments. The actual assignment results depend on the values of the fit and fill coefficients, which cannot be set at the ideal levels without understanding the impact on solution quality. So, analyzing run results to get insight and develop a “good” solution is now discussed.

Four measures are considered particularly useful in evaluating alternative solutions.

1. Fitness premium. The Marine-to-school fitness function is defined in a way that for each school, the average fitness over all Marines *eligible* for that school is 100. So, for any school, if the average fitness computed over Marines *assigned*

to that school is greater than (or less than) 100, the difference represents a positive (or negative) premium.

2. Unfilled seats in first week. Since the model is run nearly every week, seats left unfilled in classes starting in the first week will remain unfilled. This represents wasted resources.
3. Average wait time for Marines. Experience has shown that the longer a Marine waits before attending their ELS, the greater the chance of disciplinary problems. Therefore, it is generally better to keep this wait period short.
4. Unassigned Marines. A purpose of the system is to assign each Marine to a school. Therefore, knowing the number of unassigned Marines is an important measure.

To provide the RDdss manager insight into the assignment results, these measures have been incorporated into output representations, including summary statistics and detailed graphs. Each output provides both numerical and visual information. The numbers are used for quantitative comparisons. The graphical information provides a big picture view for use in a qualitative comparison.

As an example, consider the two fitness graphs in Figure 16. Both of these were produced using an RD1 file of 344 Marines and an BNA extract with school data covering a 120 day period. Below each graph is amplifying and numerical information. The bar chart on the left has a fit coefficient of 0 and a fill coefficient of 1. Additionally, its fitness premium is 28 ($=128-100$). The bar chart on the right has a coefficient of 1 for both fit and fill. Its fitness premium is 38 ($=138-100$). The fit numbers are quantitatively comparable. The fitness premium for the run on the right is 36% ($=100\% \times [38-28]/28$) better than the run on the left.

The above numerical analysis is also supported by visually comparing the bar charts. Notice the large number of blips falling below the fitness score of 100 on the left graph. Each blip represents a Marine, and there are actually 44 of them. Conversely, notice the comparatively small number of blips below 100 on the right graph. There are 21 of them. The visual comparison of the two graphs reveals the

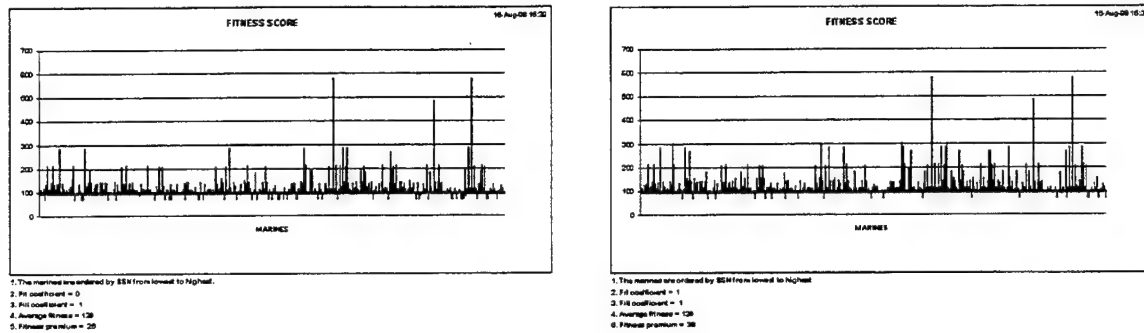


Figure 16. Two alternative results: Fit

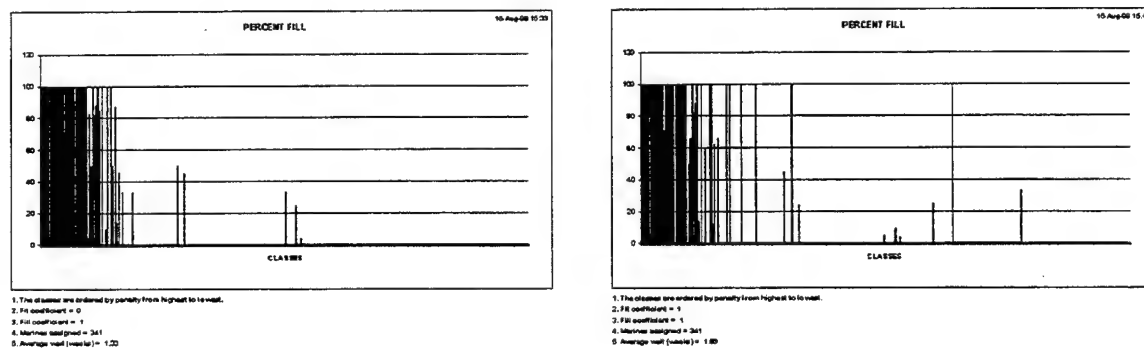


Figure 17. Two alternative results: Fill

qualitative difference. The graph on the right indicates a higher quality Marine-to-school fit. Of course, this supports the numerical comparison performed earlier.

As another example, consider the two fill graphs in Figure 17. These are based on the two model runs just described. In other words, the left graph has a fit coefficient of 0 and a fill coefficient of 1, and the right graph has a coefficient of 1 for both fit and fill.

Visually, these two graphs display an obvious difference. To appreciate how they are different requires an understanding of the significance of the blips. As note 1 of each graph indicates, "The classes are ordered by penalty from highest to lowest." Recall from Chapter III, this means the ordering is by class report date, starting with the earliest date. In other words, the bar chart on the left indicates almost every Marine will start school within the first three weeks. As discussed earlier, this is desirable. The bar chart on the right has its blips more spread out. So, by visually

comparing the two graphs, the left one has a higher quality with respect to minimizing wait time.

This is easily validated by the numbers given in note 5 of each graph. The average wait of a Marine from the left bar chart is 1.33 weeks, and the average wait from the other bar chart is 1.69 weeks. Now we quantitatively compare these values. The results from the left graph are 27% ($=100\% \times [1.69-1.33]/1.33$) better than the right.

In the following table, the results from three runs are given. The measures on the left are the four measures described earlier. Each run used the same 344 Marines and 120 days worth of school classes. The information for the first and third runs is from the two model runs discussed above. The second run was determined by “gaming” the system. In other words, we tried numerous combinations of the fit and fill coefficients until a “good” solution was found. Run 2 was the result. Understand though, we are not saying “best” solution. Only the USMC can determine what is best for them.

	Run 1 (fit=0; fill=1)	Run 2 (fit=1; fill=11)	Run 3 (fit=1; fill=1)
Unfilled Seats (1st week)	23	23	23
Unassigned Marines	3	3	3
Average Wait (weeks)	1.33	1.33	1.69
Fitness Premium	28	31	38

The reason we feel Run 2 is a “good” solution is based on a quantitative comparison of the measures and one additional piece of information. It is the fact the USMC feels it is more important to minimize wait time than to maximize the Marine-to-school fit. The remainder of this section explains how we decided Run 2 was a “good” solution.

Starting with the values for unfilled seats, it is obvious all three runs are equal. The same is true for the unassigned Marines measure. So, these two measures provide no insight for comparison purposes.

Concerning the average wait, the values of Runs 1 and 2 are equal. Further, their 1.33 value is numerically smaller than the 1.69 value of Run 3. This is good for Runs 1 and 2. As was proven earlier, a value of 1.33 is 27% better than 1.69.

Finally, there is the fitness premium measure. The first observation made is that Run 1 is a candidate for elimination. The reason is as follows. Run 2 has the same score as Run 1 for the first three measures in the table, but it shows an 11% ($=100\% \times [31-28]/28$) increase in the fitness premium. So, Run 1 is eliminated from the competition.

This leaves the last two runs. Run 3 shows a 23% ($=100\% \times [38-31]/31$) increase in fitness premium over Run 2. This creates a conflict. Run 3 has a 27% increase in average wait time, which is bad. Conversely, it has a 23% increase in fitness premium, which is good. To decide which solution is better, more information is necessary. This comes from the fact mentioned earlier, that the USMC feels it is more important to minimize wait time than to maximize the Marine-to-school fit.

We can now conclude Run 2 is a better solution than Run 3. Here is why. Run 2 minimizes average wait by 27% by giving up a 23% increase in Marine-to-school fit. This agrees with the USMC desire to more strongly emphasize a minimum waiting time. So, Run 2 is a better solution than Run 3. Since it also satisfies the known desires of the USMC, we feel it is a "good" solution as well.

E. OBJECTIVE COMPARISON OF OLD AND NEW SYSTEM SOLUTIONS

In the previous section, we described how a set of solutions generated by the RDdss are compared to determine a "good" solution. Using the same approach, we now demonstrate how to objectively compare the results from the old system with the new system. This provides the USMC a means of determining which system's solution is better, if either. Comparison of the new and old system results has been made easy by automating the importing of the RDM result into the RDdss.

We have summarized the important data for the old and new systems in the following table, based on the same 344 Marines and 120 days of school classes discussed earlier. Since the assignment algorithm for the RDM was designed to maximize fill, a run from the RDdss based strictly on fill has also been included. It is the run with a fit coefficient of 0, and a fill coefficient of 1. The last column, labeled "Run 2," is a "good" solution we found by gaming the RDdss.

	Run 1 (fit=0; fill=1)	RDM	Run 2 (fit=1; fill=10)
Unfilled Seats (1st week)	23	23	23
Unassigned Marines	3	3	3
Average Wait (weeks)	1.33	1.34	1.34
Fitness Premium	28	29	32

We start by objectively comparing the RDdss fill-based result with the RDM result. They are labeled "Run 1" and "RDM," respectively. As we shall prove, the RDdss has done a better job at filling early school seats than the RDM.

Of the four measures, only one provides any useful insight for this comparison. It is the average wait measure. The reason the other three do not apply is as follows. The first two measures have the same values, so they provide no insight. The last measure is not too meaningful, since Run 1 is strictly based on maximizing fill. In other words, it disregards fit because its fit coefficient is set to 0. This leaves the average wait measure.

A quantitative comparison of the average wait shows the RDdss solution is 1% ($100\% \times [1.34 - 1.33] / 1.33$) better than the RDM solution. This is explainable by considering the approach used in each model. The RDM assignment algorithm maximizes the fill of early school seats by looking at each school in a sequential fashion. In contrast, the RDdss conducts a global comparison of Marines and schools. In other words, it goes through numerous iterations until it finds an optimal fill solution. This shows that the RDM assignment algorithm is only closely approximating the optimized assignment result.

Finally, we "gamed" the new system until a "good" solution was found. This is Run 2. It has a fit coefficient of 1 and a fill coefficient of 10.

The objective comparison of the RDM and Run 2 solutions is straightforward. Since the first three measures are the same for both results, they provide no insights. This leaves only the fitness premium measure. A quantitative comparison of this measure shows Run 2 has a 10% ($100\% \times [32-29]/29$) better solution than RDM. This shows the benefit of "gaming" the RDdss.

F. SUMMARY

This relatively long chapter covered some important points related specifically to the RDdss. The architectural components of the new system were discussed first, followed by the steps taken to actually build this DSS. Then, from an operational point of view, using the RDdss was discussed. This included setting up a run, executing the model, and customizing a run for determining a "good" solution. Next, two innovations were discussed in great detail. These included the date change innovation and the analyzing model results innovation. Finally, a means of objectively comparing the solution from the old and new systems was discussed.

We truly believe the innovations and capabilities introduced by this prototype can greatly benefit the USMC's Marine Enlisted Assignments branch. It can help the USMC to more effectively accomplish its mission as stated by the Commandant of the Marine Corp, "to put the right Marine in the right place at the right time with the right skills and quality of life."

V. CONCLUSION

This concluding chapter has three short sections, followed by a final comment. In the first section, we reiterate the significant contributions made in the thesis. This is followed by a brief description of some lessons we learned. Then, some recommended improvements for the prototype are given. Following this last section, we comment on how this thesis work might benefit the other branches of the United States military.

A. SIGNIFICANT CONTRIBUTIONS

Four significant contributions beneficial to the USMC were made in this thesis. These were discussed in detail in the previous three chapters. A list and summary of the contributions is given here.

- Analysis and articulation of the recruit distribution process - by interviewing USMC and DSAI personnel, reviewing available documentation, and operating the RDM we were able to articulate an understanding of the recruit distribution process using IDEF process modeling. Additional data modeling was articulated in a third normal form relational schema.
- Development of a mathematical model - by analyzing the assignment process, criteria, and constraints, USMC policies and objectives, a mathematical model for the new system was developed.
- Fully functional prototype - an intuitive and easy to understand new system that seamlessly interfaces with the old system was built. It provides an objective means of comparing assignment results of both systems, and was developed using COTS software applications.
- Elimination of the proprietary solver and associated contractor lock-in - this was accomplished by replacing the proprietary solver with two COTS applications.

B. LESSONS LEARNED

Our original goal was to build an As-Is IDEF0 model of the RDM, about two months after starting the thesis. This worked out fine. The next goal was to take

another six weeks to develop a To-Be IDEF0 model [Ref. 1]. This did not work out, for two reasons. First, our understanding of the entire recruit distribution process was still not mature enough to build the To-Be model. Nearly half a year was spent just thinking about its many details. Second, it is nearly impossible to envision the new system without first knowing the capabilities and limitations of the application used to build the prototype. One extreme envisions the impossible, while the other barely touches the reengineered system's potential. *The lesson learned from this was that properly reengineering a process requires two things. A detailed understanding of the current process environment, and a good working knowledge of the capabilities and limitations of the application used to build the prototype.*

Getting feedback from the Marine Enlisted Assignments branch was not always easy. Correspondence normally occurred by e-mail. There was an inverse relationship between the number of questions asked and the number of answers given. As the one increased, the other decreased. *So, another lesson learned was that it is best to limit each correspondence to a question or two.*

While writing code, it is best to simultaneously document the code's purpose. Going back and trying to provide documentation at the end is too frustrating. This requires a considerable amount of discipline, but is well worth the effort. *The lesson learned here is document as you code.*

C. PROTOTYPE IMPROVEMENTS

A big part of this thesis involved the building and refinement of the prototype. The following topics are areas where the RDdss could use improvements.

1. Speed Improvements

a. Preprocessing Data

Of all the steps necessary to complete a run of the RDdss, the preprocessing step is the most time consuming. For the 344 Marines and approximately 500 school classes analyzed in this thesis, it takes about 30 minutes to preprocess

this data. This is about ten times longer than the assignment model takes to find an optimal solution. As the size of the preprocessing data is doubled, so is the time needed to preprocess it. Finding a way to speed up this process is desirable. One possibility is to find a way to perform all the preprocessing in main memory, with only two accesses to the hard drive. The first access is to get the necessary Marine and school data. The second is to write the preprocessed results back to the hard drive.

b. Analyzing Result

Compared to all other forms and switchboards in the RDdss, opening the "analyze results" forms is the most time intensive. With the 344 Marines and approximately 500 school classes, it takes just under a minute for any of these three to open. This feels like an eternity when you are in a hurry. The reason it takes so long is the creation of temporary tables, numerous calculations and comparisons, and retrieving of data from flat files. Finding a way to reduce this time is desirable. One possibility is to create another form which would contain only the four objective measures. Currently, the RDdss manager must go to two different forms for this information. Each containing information and displays in addition to the objective measures, which slow them down.

2. Multiple Solution Storage

Currently, the prototype will only save the results of one solution. This means the RDdss manager must either print-out hard copies of the graphs, or manually write down each solution's objective measures for comparison with other runs. An improvement here should allow the storage of at least three solutions. This would permit the RDdss manager to easily compare each solution. He or she could then eliminate one or two undesirable solutions, while continuing to search for a "good" one.

3. Administration Switchboard

To look at all the schools or properties in one display, the RDdss manager must open up the associated school or property table. A reason for doing this is system administration. For instance, if a supervisor is auditing the RDdss's property data, looking at all the data at one time is convenient. A simple way of including this feature is by adding another switchboard. The switchboard's buttons would have VBA code associated with them, causing the desired tables to open.

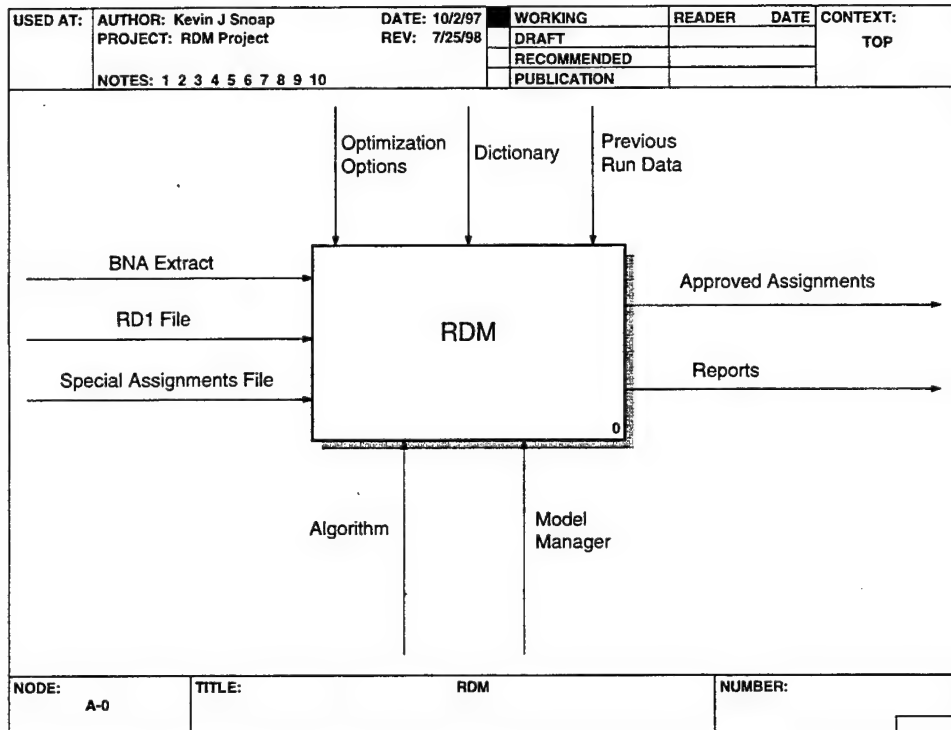
Any benefits derived from this thesis work were specifically aimed at the USMC. However, it does not have to end there. Like the USMC, other branches of the United States military must deal with the problem of how best to assign their service members to schools. Since all the services require their enlisted personnel to take the ASVAB test, the approach used in our prototype might prove an ideal solution. Further, our efforts may also contribute to the reengineering of other assignment models. So, this thesis work could actually benefit all branches of the United States military.

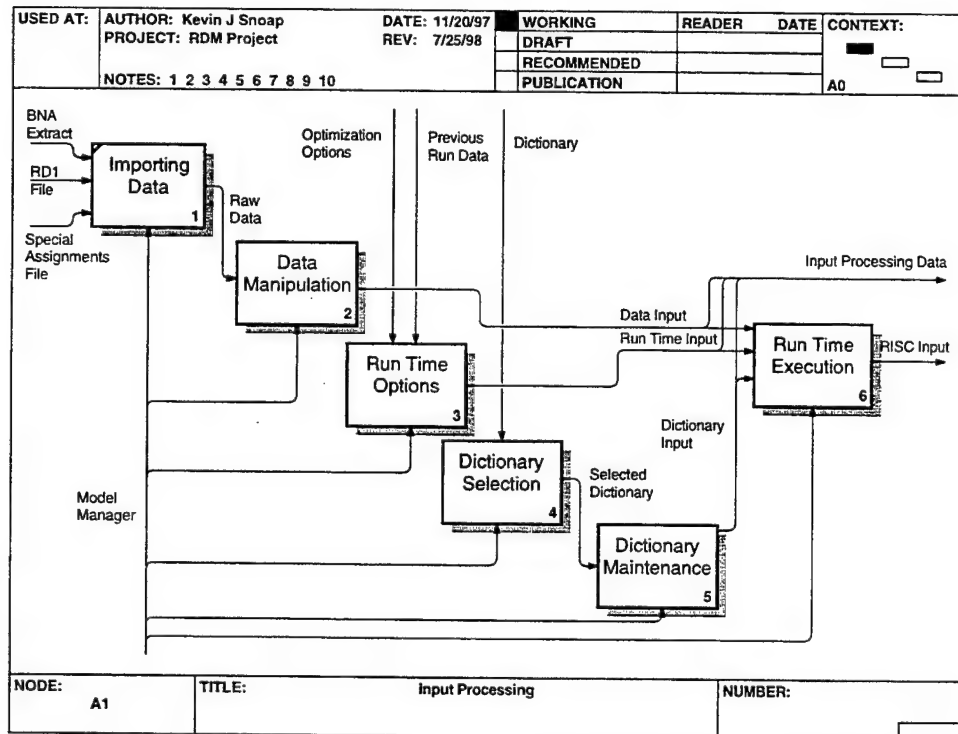
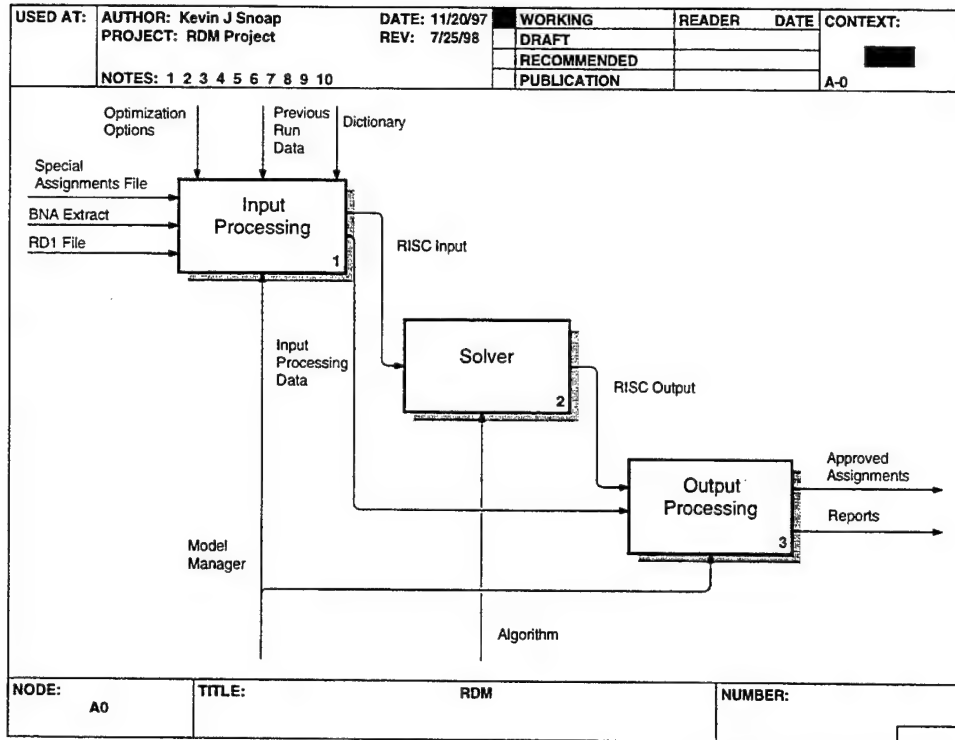
APPENDIX A. ACRONYMS

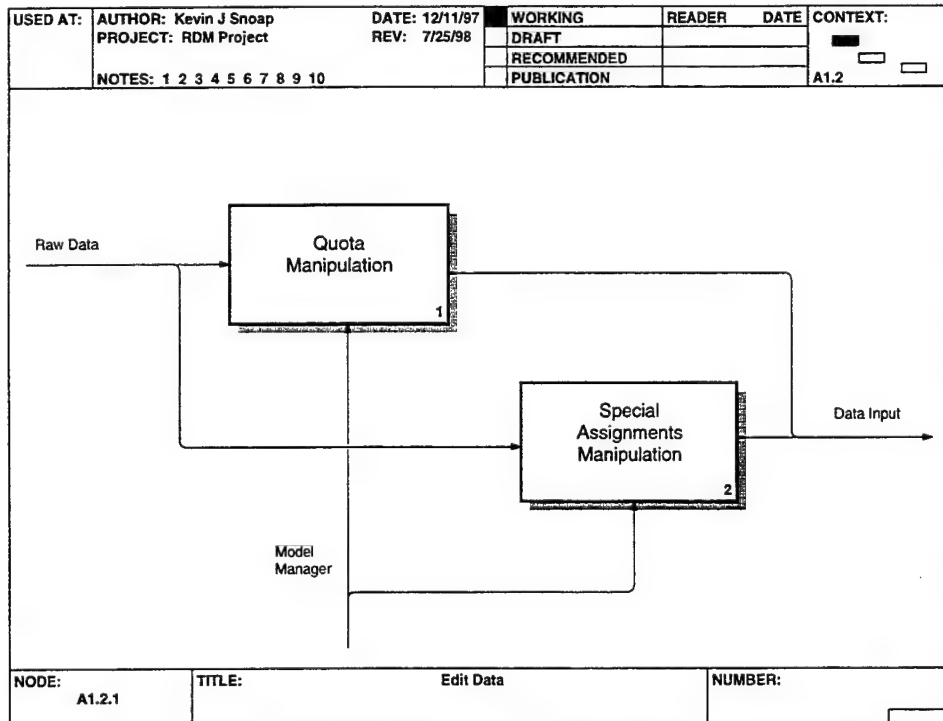
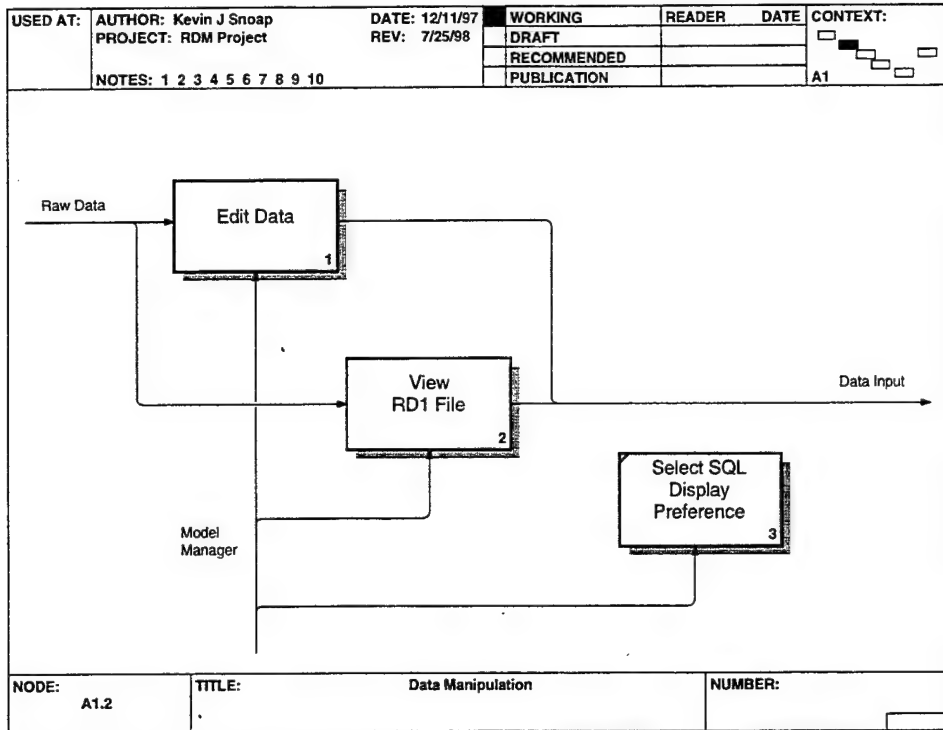
Acronym=>Meaning

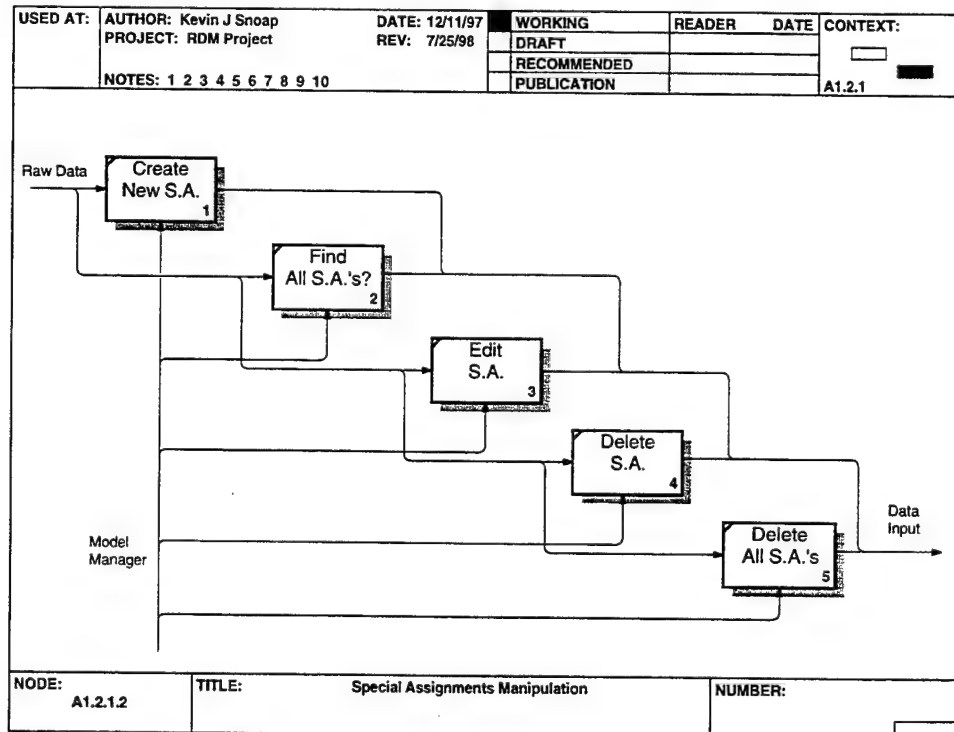
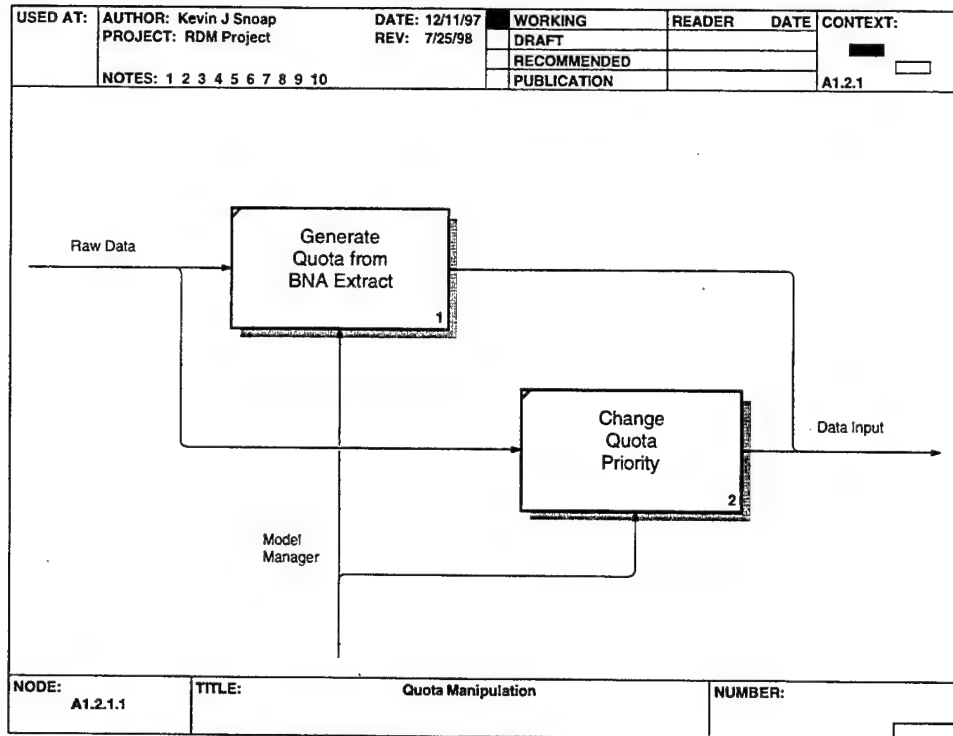
AMPL	A Modeling Language for Mathematical Programming
ARMS	Automated Recruit Management System
ASVAB	Armed Services Vocational Aptitude Battery
BNA	By Name Assignment System
BPWin	Business Process for Windows
COTS	Commercial-off-the-shelf
CPlex	Optimization Package for Complex Linear, Network and Integer Programming
DSAI	Decision Support Associates, Inc.
DSS	Decision Support System
ELS	Entry Level School
ICOM	Input, Control, Output, Mechanism
IDEF	Defense Institute Modeling
IDEF0	Business Process Modeling
MCRD	Marine Corps Recruit Depot
MCT	Marine Combat Training
MODS	Military Operational Data Store
MOS	Military Occupational Specialty
PEF	Program Enlisted For
RD1	Recruit Distribution Input File
RD3	Recruit Distribution Output File
RDdss	Recruit Distribution Decision Support System
RDM	Recruit Distribution Model
USMC	United States Marine Corps
VBA	Visual Basic for Applications

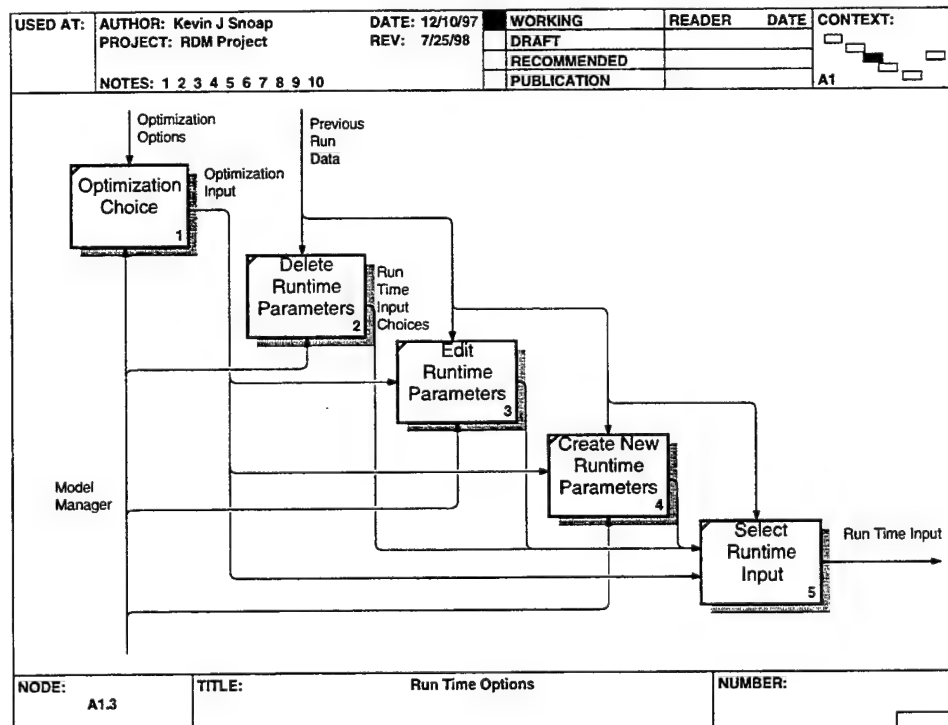
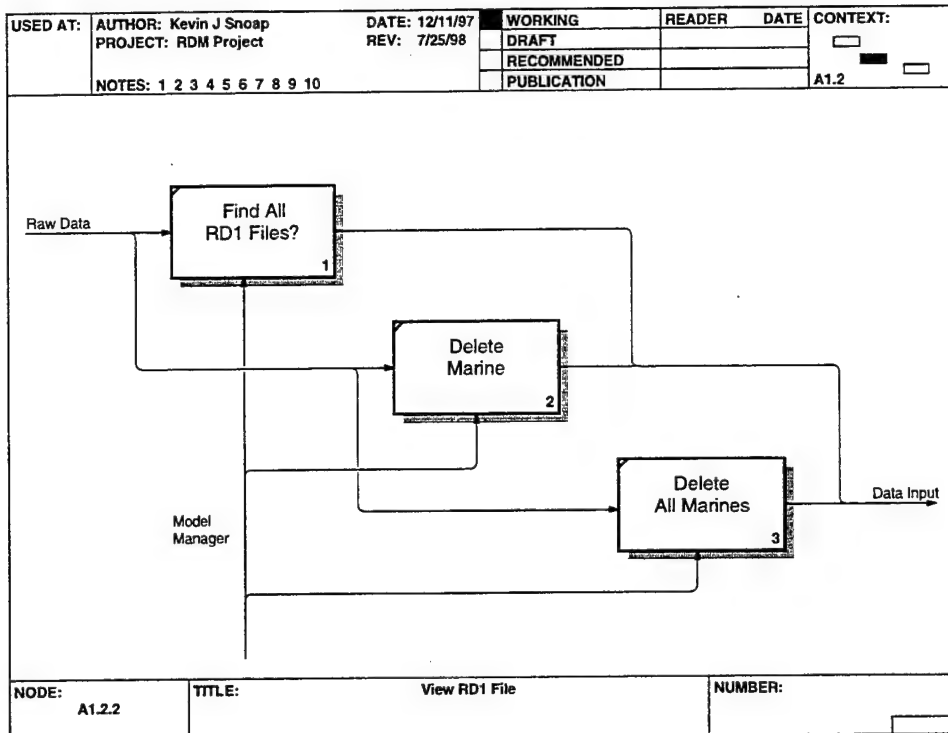
APPENDIX B. AS-IS RD BUSINESS PROCESS (IDEF0) MODEL

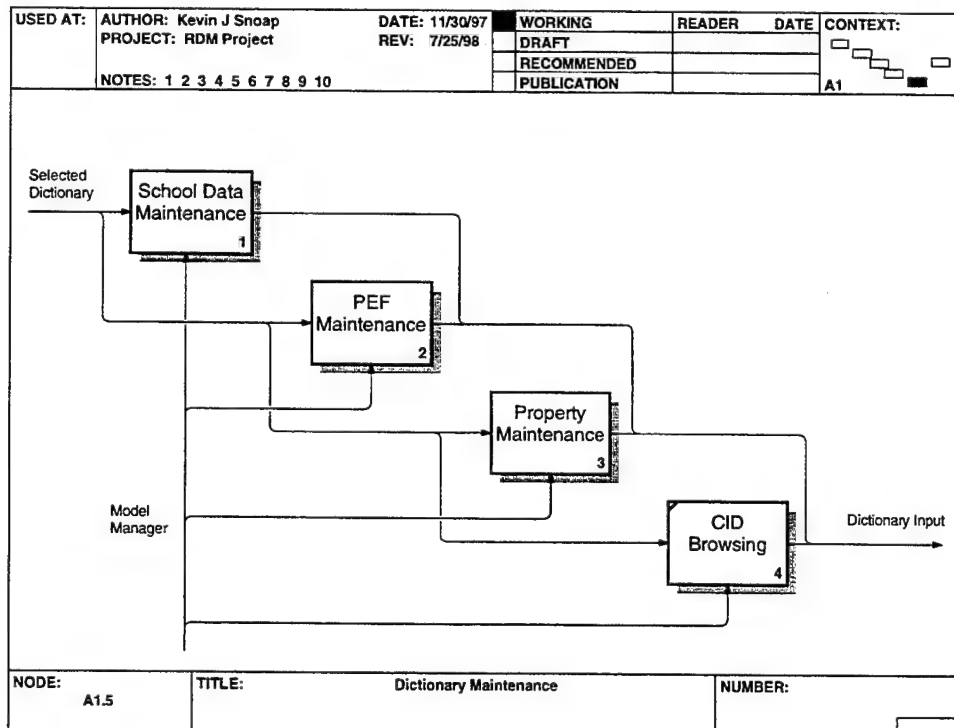
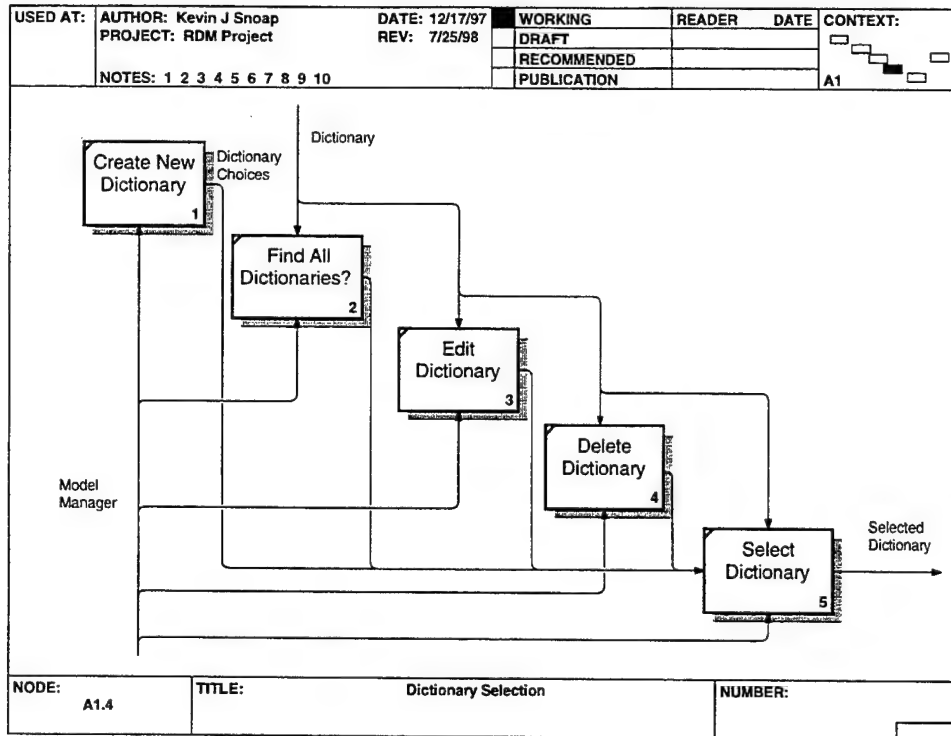


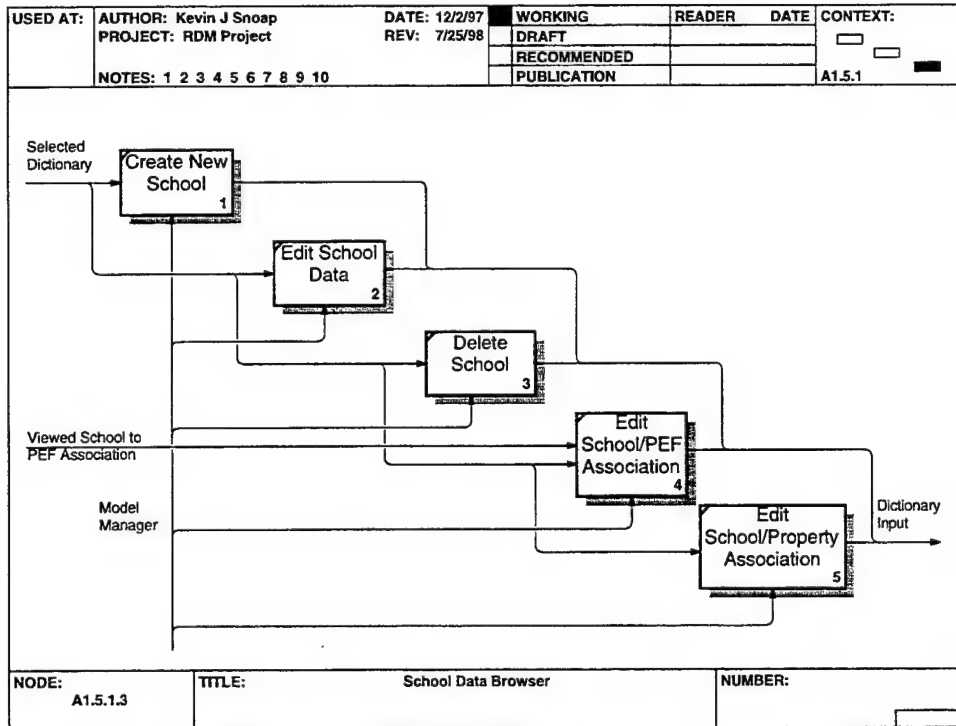
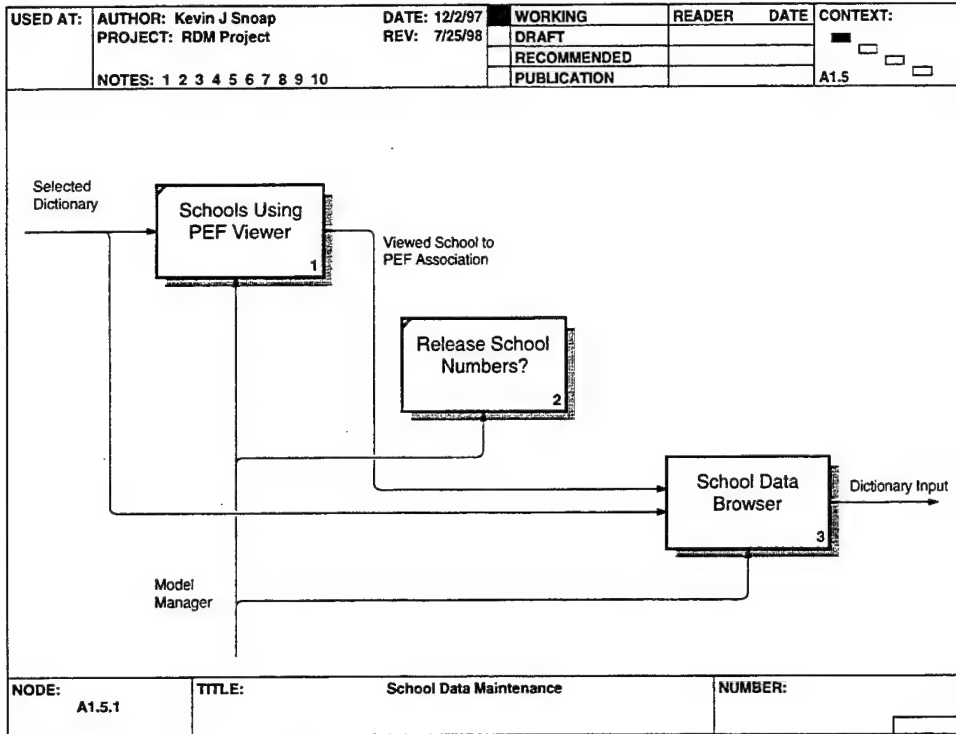


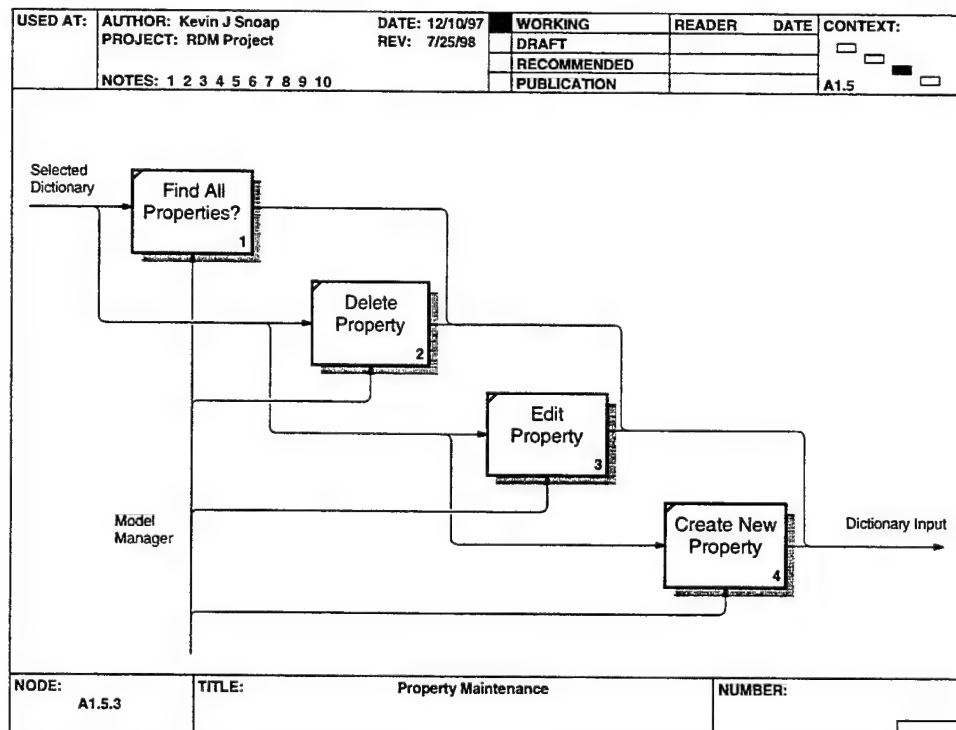
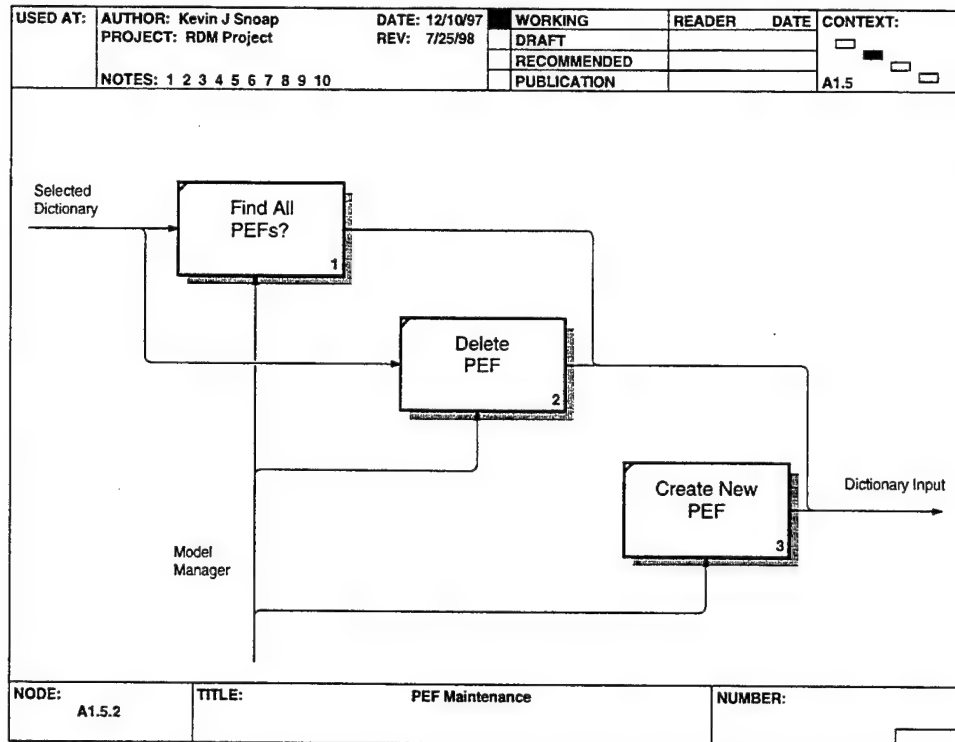


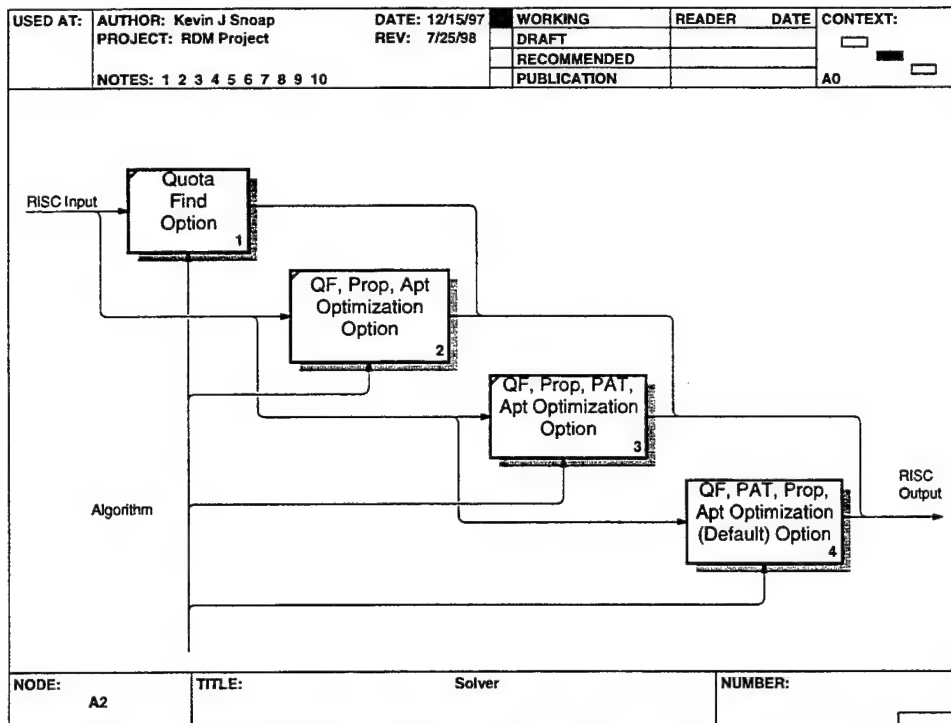
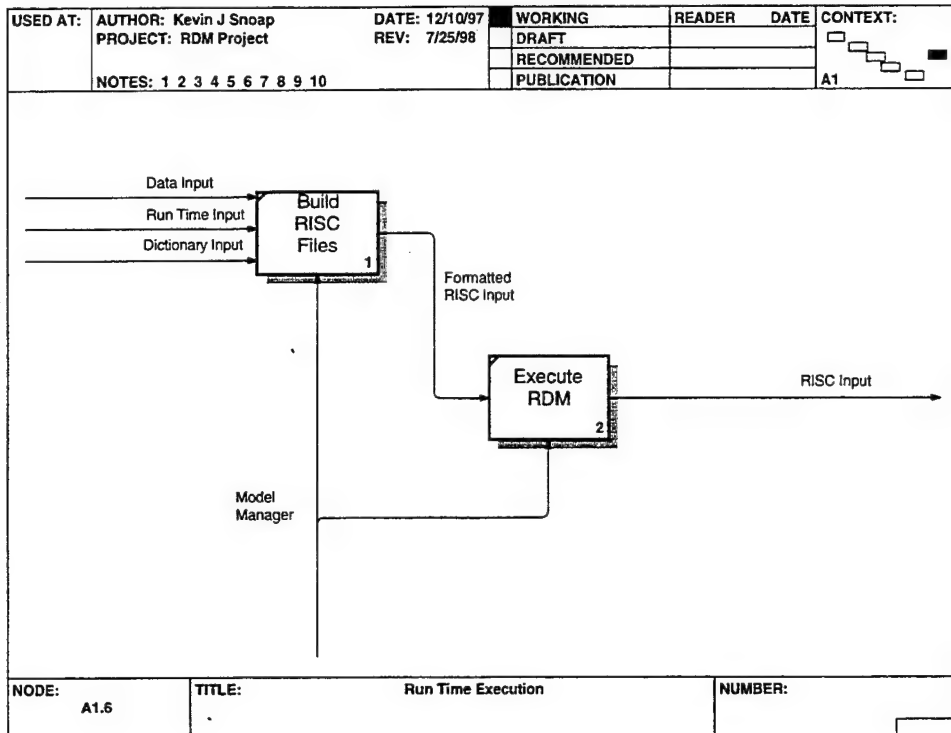


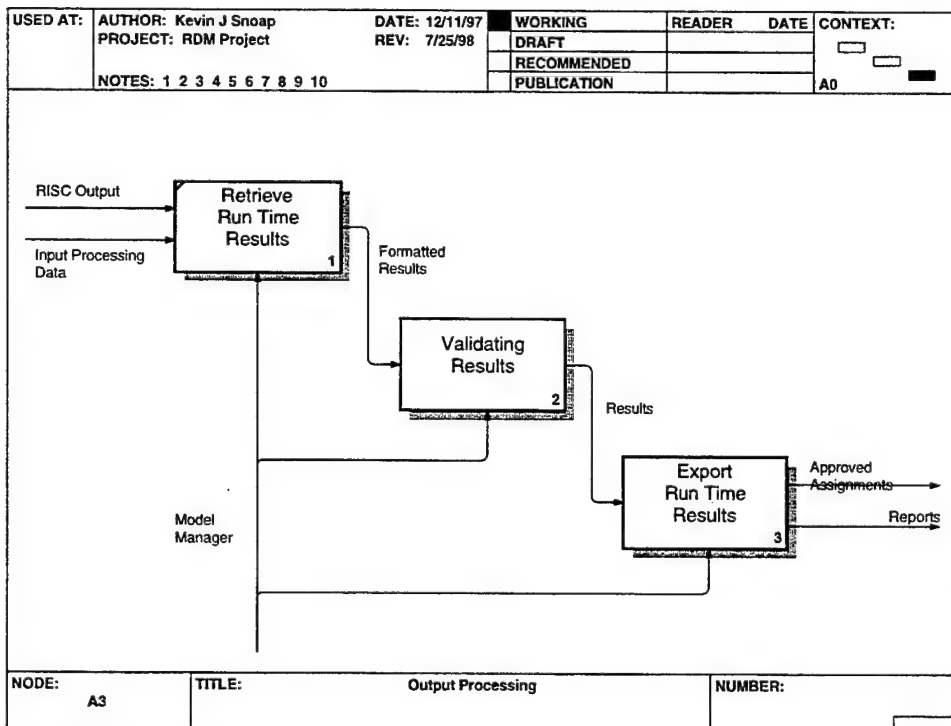
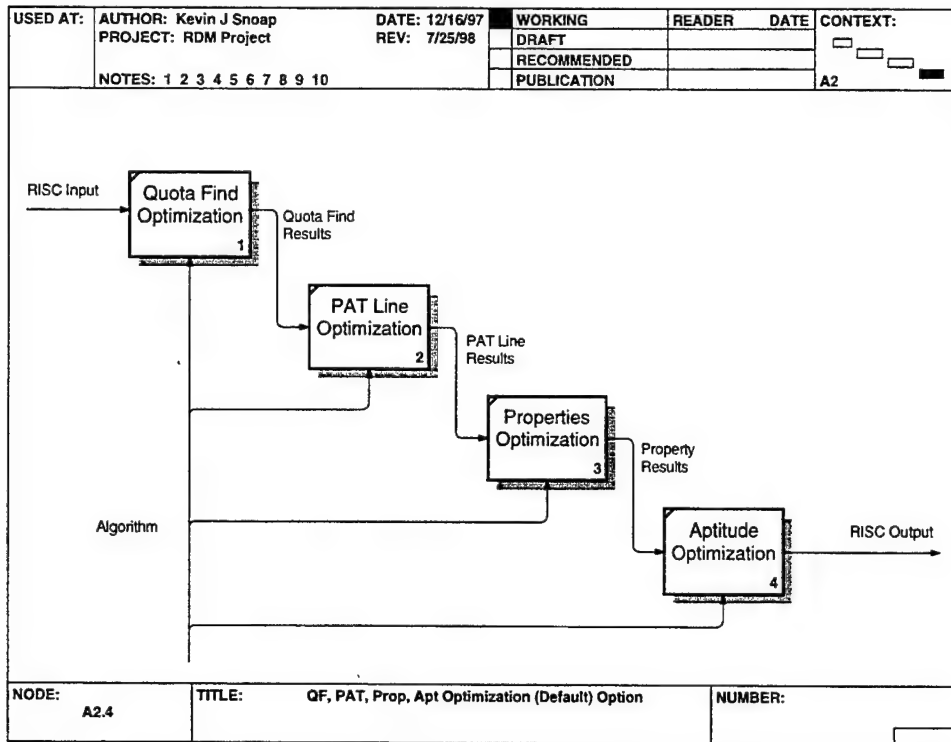


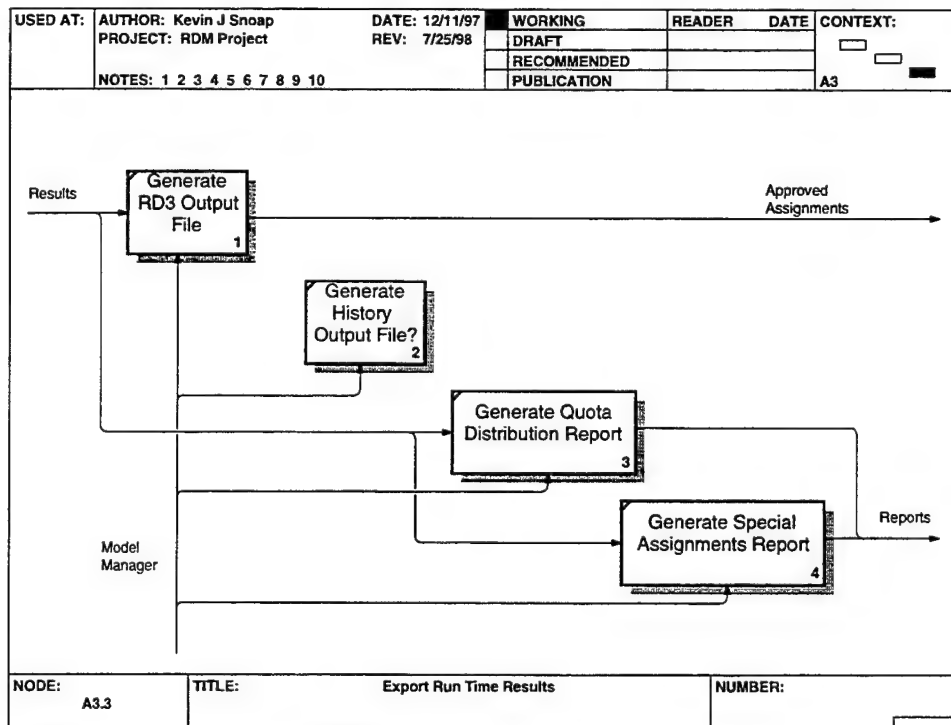
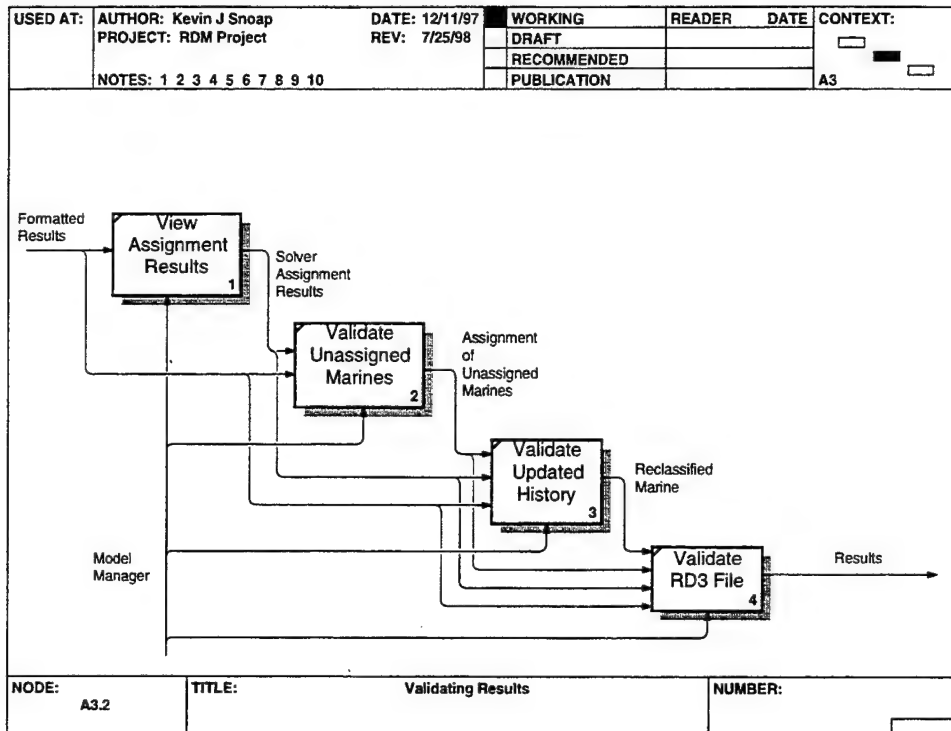




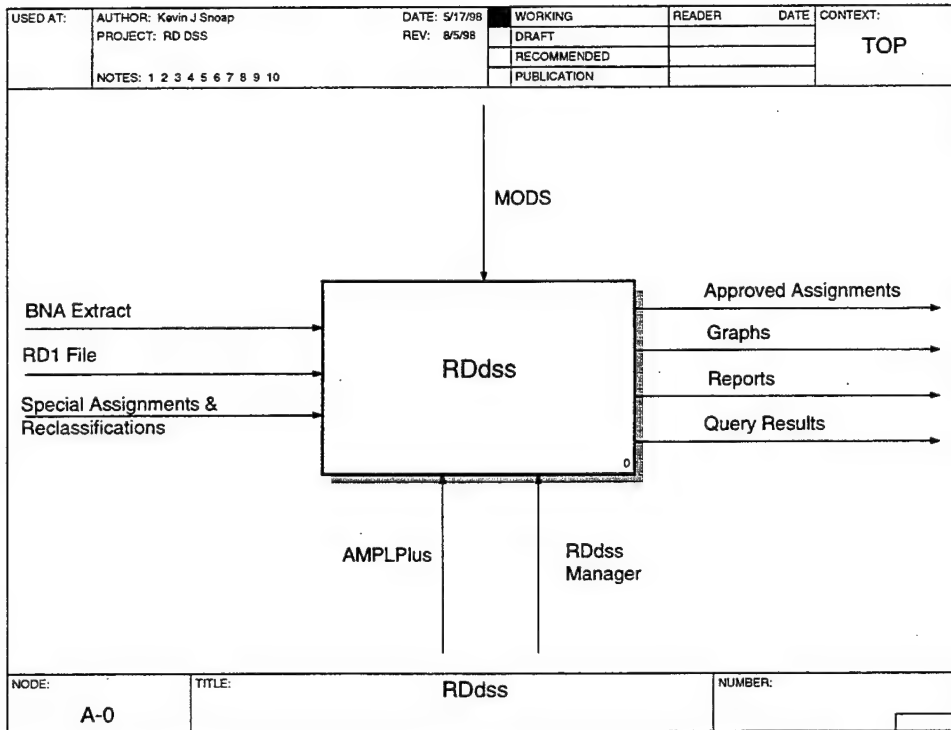


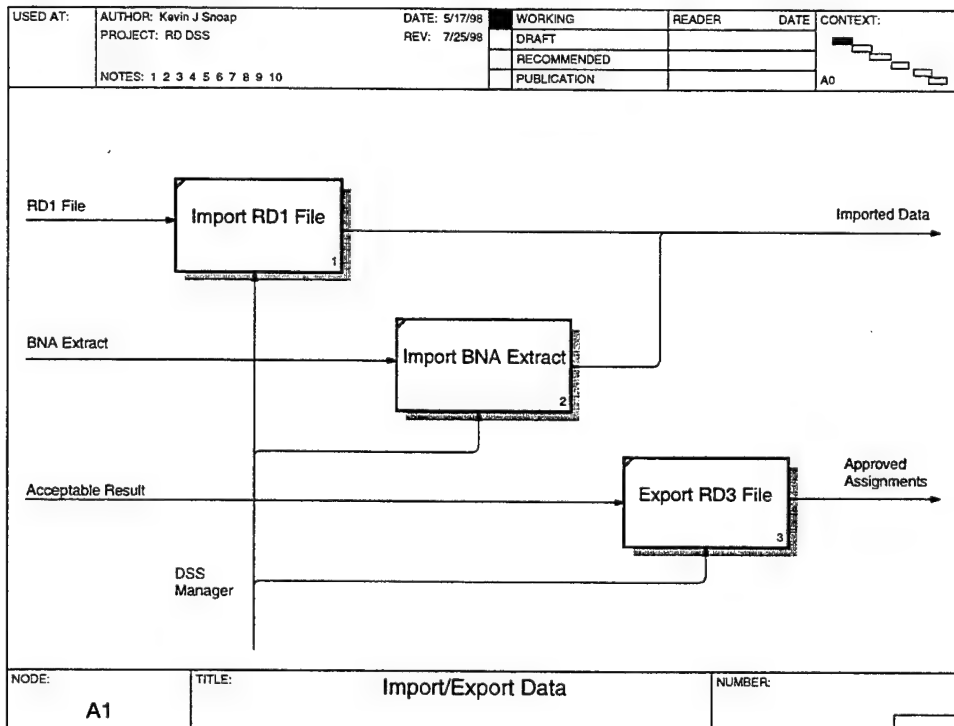
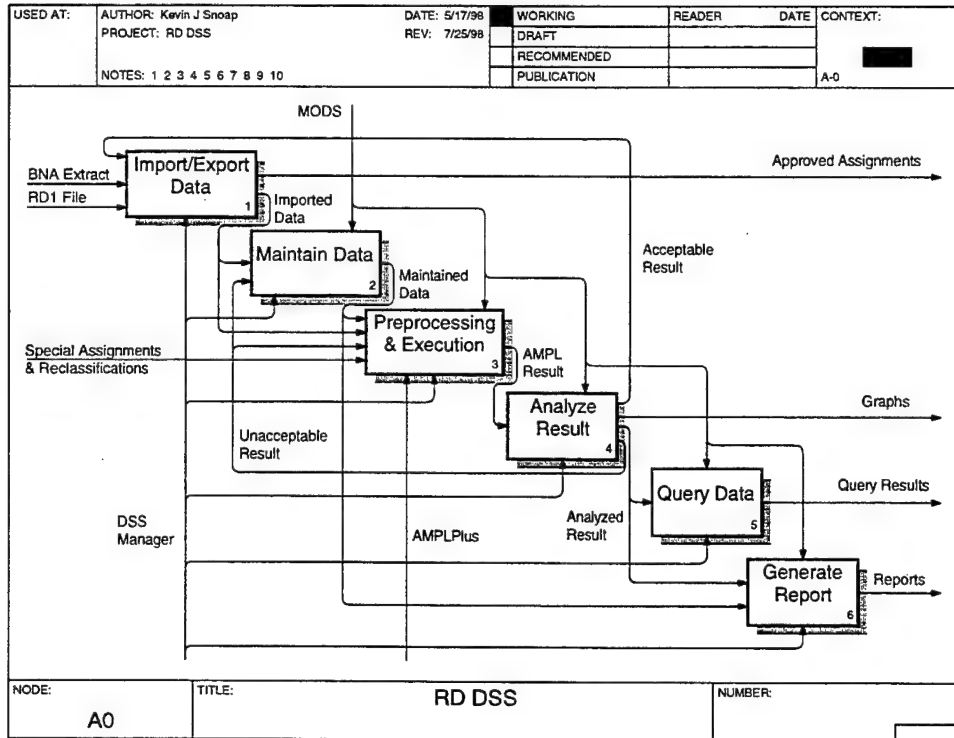


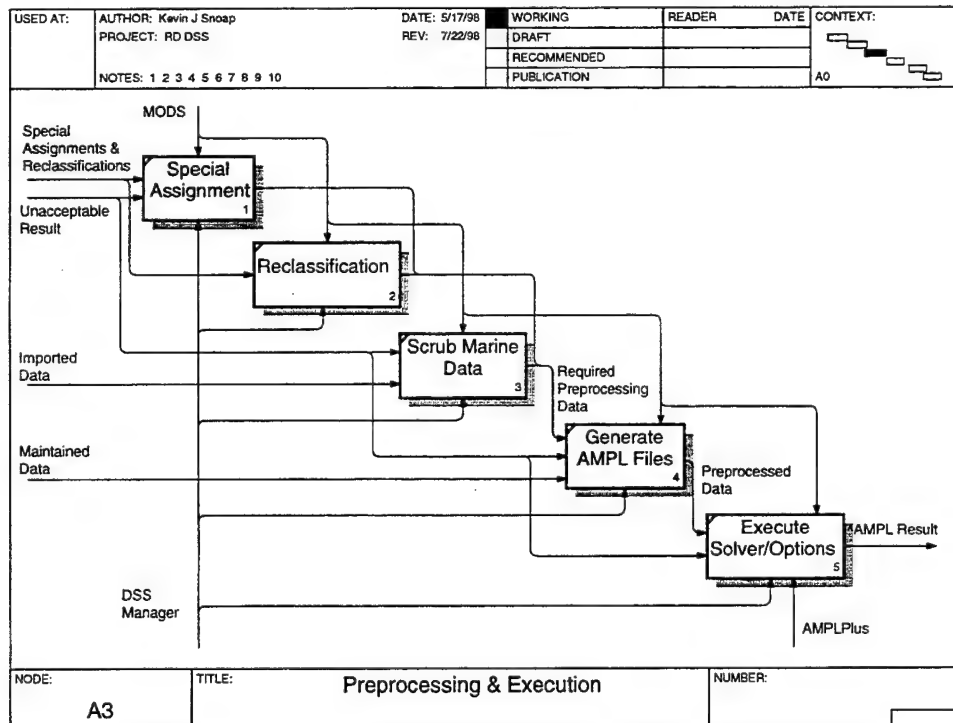
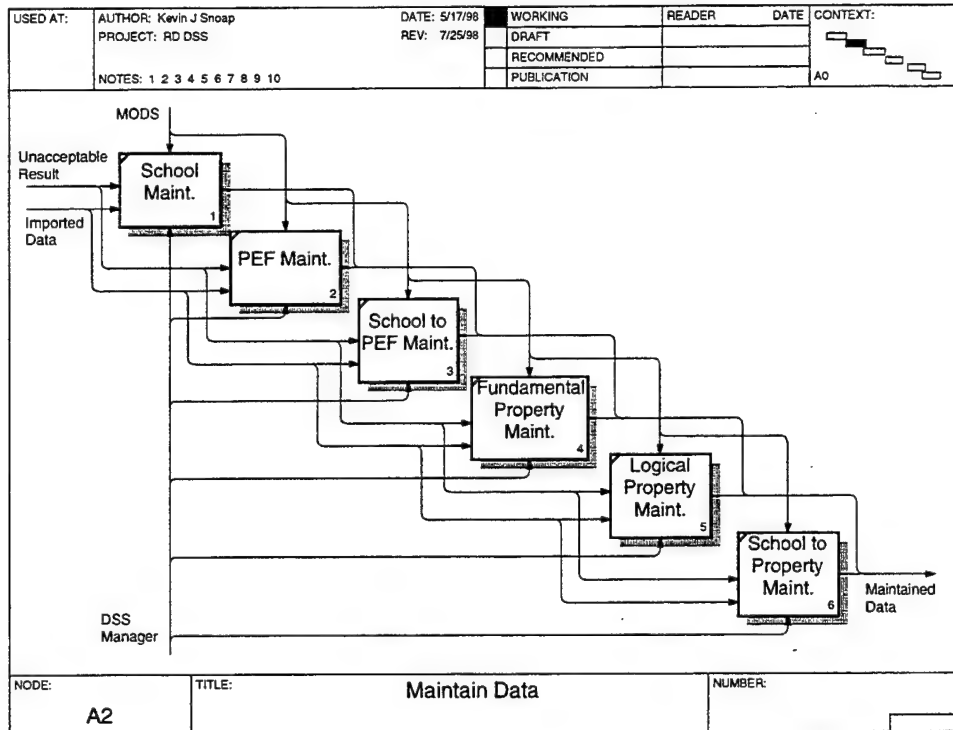


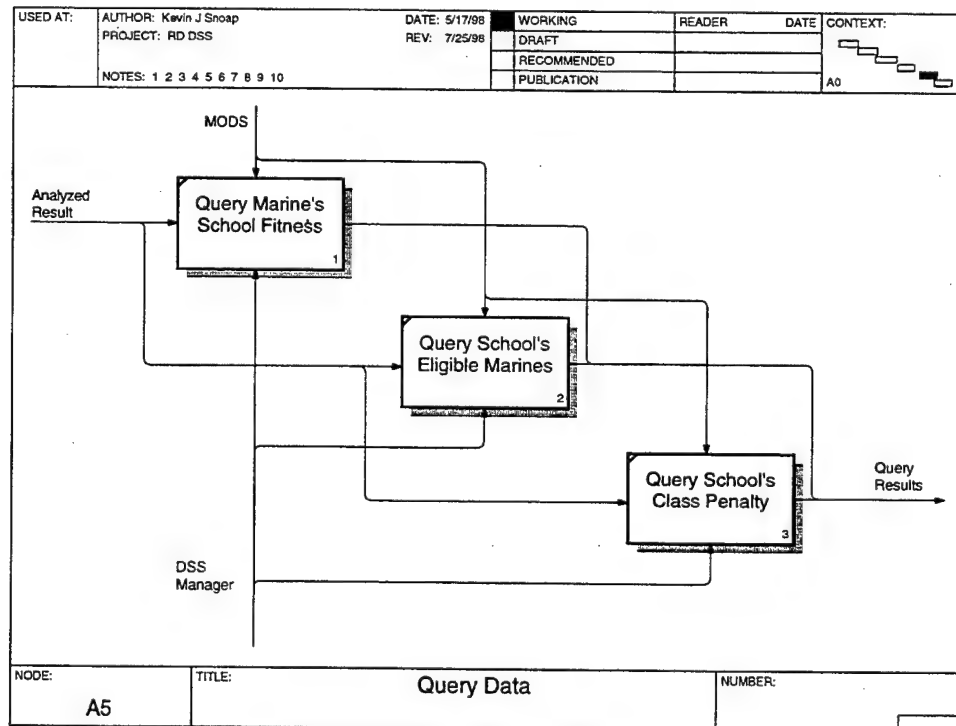
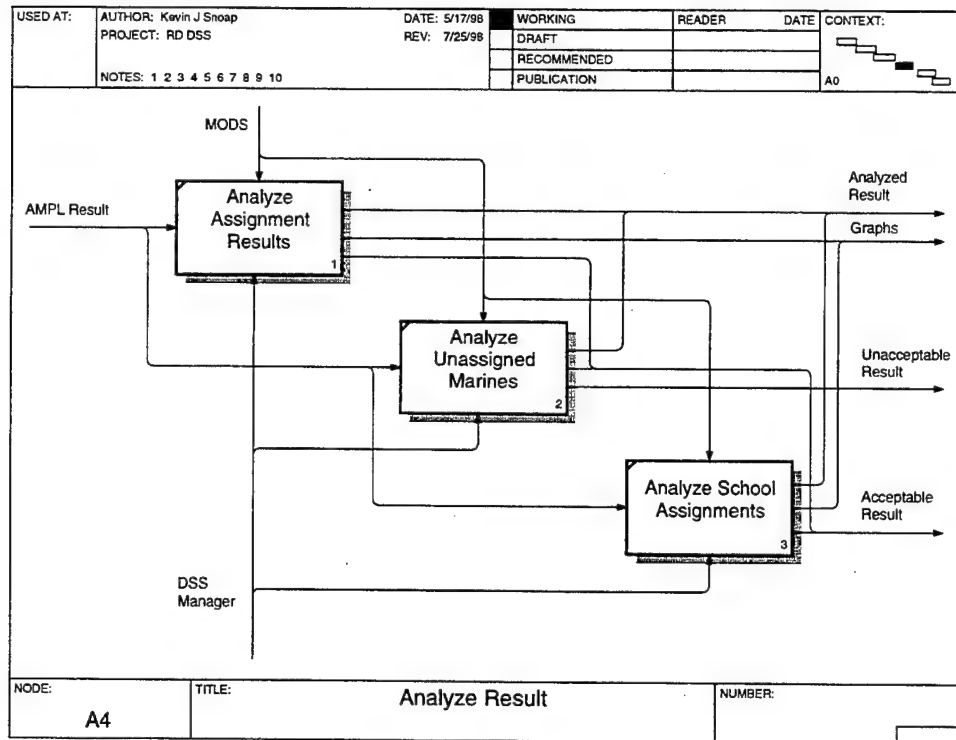


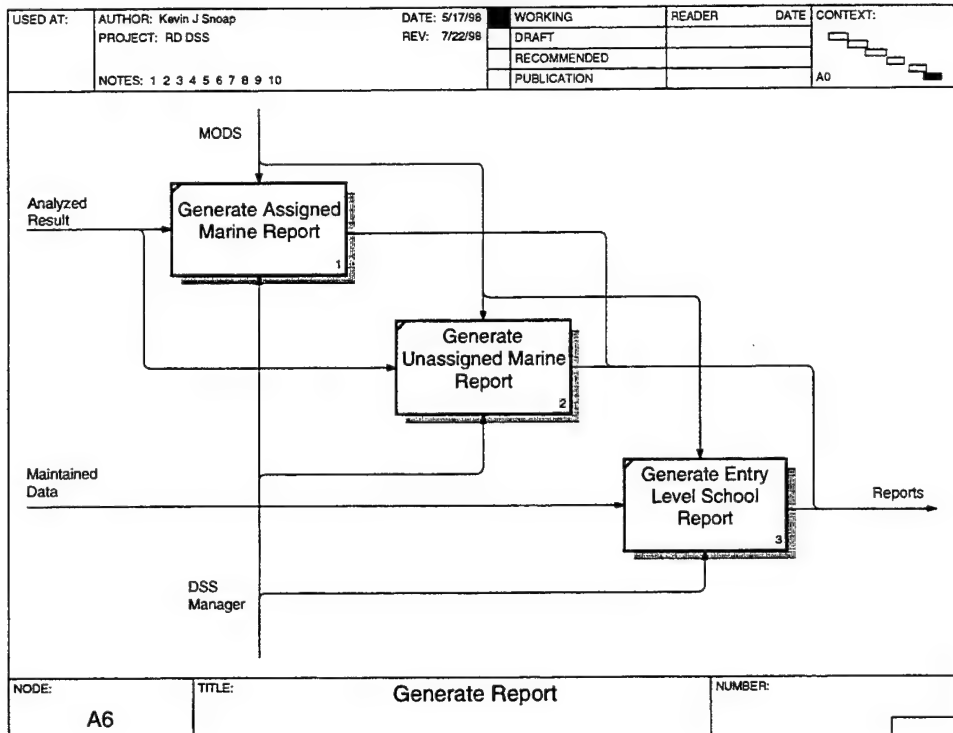
APPENDIX C. TO-BE RD BUSINESS PROCESS (IDEF0) MODEL











APPENDIX D. RDDSS VBA CODE

```

1 Form: frmAnalyzeResult
2 Code
3 1 Variable VB_Name = "Form_frmAnalyzeResult"
4 2 Attribute VB_Creatable = True
5 3 Attribute VB_PredeclaredId = True
6 4 Attribute VB_Exposed = False
7 5 Option Compare Database
8 6 Option Explicit
9
10 Dim booSchoolAssignment As Boolean
11
12
13 Private Sub btnExit_Click()
14 On Error GoTo Err_btnExit_Click
15
16
17 DoCmd.Close
18
19 Exit_btnExit_Click:
20 Exit Sub
21
22 Err_btnExit_Click:
23 MsgBox Err.Description
24 Resume Exit_btnExit_Click
25
26 End Sub
27
28
29 Private Sub btnSchoolAssignments_Click()
30 On Error GoTo Err_btnSchoolAssignments_Click
31
32 Dim stDocName As String
33 Dim stLinkCriteria As String
34
35 'Set the flag
36 booSchoolAssignment = True
37
38 'Close the current form
39 DoCmd.Close
40
41 'Open specified form
42 stDocName = "frmSchoolAssignments"
43 DoCmd.OpenForm stDocName, , stLinkCriteria
44
45 Exit_btnSchoolAssignments_Click:
46 Exit Sub
47
48 Err_btnSchoolAssignments_Click:
49 MsgBox Err.Description
50 Resume Exit_btnSchoolAssignments_Click
51
52 End Sub
53
54 Private Sub btnSolverAssignments_Click()
55 On Error GoTo Err_btnSolverAssignments_Click
56
57 Dim stDocName As String
58 Dim stLinkCriteria As String
59
60 'Close the current form
61 DoCmd.Close
62
63 'Open specified form
64 stDocName = "frmAssignmentsResult"
65 DoCmd.OpenForm stDocName, , stLinkCriteria
66
67 Exit_btnSolverAssignments_Click:
68 Exit Sub
69
70 Err_btnSolverAssignments_Click:
71 MsgBox Err.Description
72 Resume Exit_btnSolverAssignments_Click
73
74 End Sub
75
76 Private Sub btnUnassignedMarines_Click()
77 On Error GoTo Err_btnUnassignedMarines_Click
78
79 Dim stSQL As String
80 Dim rec As Recordset
81 Dim db1 As Database
82
83 Set db1 = CurrentDb()
84
85 'If the number of unassigned marines = 0, then disable the
86 unassigned marines button
87 stSQL = "SELECT Count(PEP) AS TotalUnassigned FROM"
88 Set rec = db1.OpenRecordset(stSQL, dbOpenSnapshot)
89 If rec.TotalUnassigned = 0 Then
90 MsgBox ("All Marines were assigned.")
91 Else
92
93 Dim stDocName As String
94 Dim stLinkCriteria As String
95
96 'Close the current form
97 DoCmd.Close
98
99 'Open specified form
100 stDocName = "frmUnassignedMarines"
101 DoCmd.OpenForm stDocName, , stLinkCriteria
102
103 End If
104
105 Exit_btnUnassignedMarines_Click:
106 Exit Sub
107
108 Err_btnUnassignedMarines_Click:
109 MsgBox Err.Description
110 Resume Exit_btnUnassignedMarines_Click
111
112 End Sub
113
114 Private Sub Command1_Click()
115 On Error GoTo Err_Command1_Click
116
117 Dim stDocName As String
118 Dim stLinkCriteria As String
119
120 'Close the current form
121 DoCmd.Close
122
123 'Open specified form
124 stDocName = "frmRDM_Main_Switchboard"
125 DoCmd.OpenForm stDocName, , stLinkCriteria
126
127 Exit_Command1_Click:
128 Exit Sub
129
130 Err_Command1_Click:
131 MsgBox Err.Description
132
133 Resume Exit_Command1_Click
134
135 End Sub
136
137 Dim stSQL As String
138 Dim rec As Recordset
139 Dim db1 As Database
140
141 Set db1 = CurrentDb()
142
143 'Create a table to get the fit and fill weights used during the
144 Application Selection "Confirm Action Queries", False
145 'Delete the table if it already exists
146 DeleteTable
147
148 db1.Execute "CREATE TABLE PERCENTAGE (CourseNumber TEXT, AMOS
149 TEXT, Quote Integer, Assigned INTEGER, Percentage INTEGER);"
150 Application.SetOption "Confirm Action Queries", True
151
152 stSQL = "SELECT * FROM PERCENTAGE;"
153 Set rec = db1.OpenRecordset(stSQL, dbOpenDynaset)
154
155 stSQL = "SELECT * FROM qryTotalQuoteForRun ORDER BY"
156 Set rec2 = db1.OpenRecordset(stSQL, dbOpenSnapshot)
157 stSQL = "SELECT * FROM qryTotalQuoteFiledForRun ORDER BY"
158 Set rec3 = db1.OpenRecordset(stSQL, dbOpenSnapshot)
159
160 'Ensure there are values in the records.
161 If rec1.EOF = False Then
162 If rec2.EOF = False Then
163 If rec3.EOF = False Then
164 rec1.MoveFirst
165 rec2.MoveFirst
166 rec3.MoveFirst
167
168 'Fill the PENALTY table with the quote values for the
169 For i = 1 To rec1.RecordCount
170 rec1.AddNew
171 rec1.CourseNumber = rec1.CourseNumber_FK
172 rec1.AMOS = rec1.AMOS_FK
173 rec1.Quote = rec1.SumOfQuote
174 rec1.Update
175 rec1.MoveNext
176 Next i
177
178 'Update the PENALTY table with the percentage of quote
179 For i = 1 To rec2.RecordCount
180 rec2.MoveFirst
181 For j = 1 To rec3.RecordCount
182 If rec2.CourseNumber = rec3.CourseNumber_FK Then
183 If rec2.AMOS = rec3.AMOS_FK Then
184 rec2.Edit
185 rec2.Assigned = rec2.CountOfRecalYear_PK
186 rec2.Percentage = Int((rec2.Assigned /
187 rec2.Quote) * 100)
188 rec2.Update
189 Exit For
190 End If
191 rec2.MoveNext
192 If i = rec2.RecordCount Then
193 rec2.Edit
194 rec2.Assigned = 0
195 rec2.Percentage = 0
196 rec2.Update
197 End If
198 rec2.MoveNext
199 Next i
200 End If
201
202 rec.Close
203 rec1.Close
204 rec2.Close
205 db1.Close
206
207 'Reset the flag
208 booSchoolAssignment = False
209
210 'Delete the PERCENTAGE table
211 DeleteTable
212
213 End If
214
215 End Sub
216
217 End Sub
218
219
220 Form: frmAssignmentResult
221 Code
222 1 Attribute VB_Name = "Form_frmAssignmentResult"
223 2 Attribute VB_Creatable = True
224 3 Attribute VB_PredeclaredId = True
225 4 Attribute VB_Exposed = False
226 5 Option Compare Database
227 6 Option Explicit
228
229 Private Sub Form_Current()
230
231 Dim stSQL As String
232 Dim rec As Recordset
233 Dim db1 As Database
234
235 Set db1 = CurrentDb()
236
237 'Calculate the high, average, and low fitness scores. Also, it
238 calculates the number of solver assignments.
239 stSQL = "SELECT Avg(Fitness) AS Average, Max(Fitness) AS Maximum,
240 Min(Fitness) AS Minimum, Count(Fitness) AS TotalAssigned FROM"
241 qryAssignmentResult;"
242 Set rec = db1.OpenRecordset(stSQL, dbOpenSnapshot)
243 Me.txtFitnessAvg = rec.Average
244 Me.txtFitnessMax = rec.Maximum
245 Me.txtFitnessMin = rec.Minimum
246 Me.txtSolverAssignments = rec.TotalAssigned
247
248 'Calculate the number of manual assignments, and the total number
249 of assignments.
250 stSQL = "SELECT Count(AssignmentType) AS TotalManualAssigned FROM"
251 qryManuallyAssignedMarines;"
252 Set rec = db1.OpenRecordset(stSQL, dbOpenSnapshot)
253 Me.txtManualAssignments = rec.TotalManualAssigned
254
255
256
257
258
259

```

```

261 Me.tblSolveAssignments
262 30
263 31 ' Calculates the number of unassigned marines, and the total number
264 of assignable marines.
265 32 strSQL = "SELECT Count(PEF) AS TotalUnassigned FROM
266 33 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
267 34 Me.tblTotalUnassigned = rec.TotalUnassigned
268 35 Me.tblTotalAssignable = rec.TotalUnassigned +
269 36
270 37 ' Create a table to get the fit and fill weights used during the run
271 38 Application.SetOption "Confirm Action Queries", False
272 39 db1.Execute "CREATE TABLE CONSTANTS (ConstantName TEXT, intValue
273 40 DoCmd.TransferText acImportFixed, "Form_Data Export Specification",
274 41 Application.SetOption "Confirm Action Queries", True
275 42
276 43 strSQL = "SELECT intValue FROM CONSTANTS;"
277 44 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
278 45 Me.tblFitWeight = rec.intValue
279 46 rec.MoveNext
280 47 rec.MoveNext
281 48 Me.tblFillWeight = rec.intValue
282 49 rec.Close
283 50
284 51 ' Remove the created table
285 52 db1.Execute "DROP TABLE CONSTANTS;"
286 53 db1.Close
287 54
288 55 End Sub
289 56 Private Sub Close_Click()
290 57 On Error GoTo Err_Close_Click
291 58 Dim stDocName As String
292 59 Dim stLinkCriteria As String
293 60
294 61 ' Close the current form
295 62 DoCmd.Close
296 63 ' Open specified form
297 64 stDocName = "frmAnalyzeResult"
298 65 DoCmd.OpenForm stDocName, , stLinkCriteria
299 66
300 67 Exit_L_Close_Click
301 68 Exit Sub
302 69
303 70 Err_Close_Click
304 71
305 72 Err_Close_Click
306 73 MsgBox Err.Description
307 74 Resume Exit_L_Close_Click
308 75
309 76 End Sub
310 77
311 78
312 Form: frmChangeGradDate
313 Code
314 1 Attribute VB_Name = "Form_frmChangeGradDate"
315 2 Attribute VB_Creatable = True
316 3 Attribute VB_PredeclaredId = True
317 4 Attribute VB_Exposed = False
318 5 Option Compare Database
319 6 Option Explicit
320 7
321 8 Private Sub btnChangeDate_Click()
322 9
323 10 ' Change the specified graduation date in the MARINE table
324 11 Application.SetOption "Confirm Action Queries", False
325 12 DoCmd.OpenQuery "qryChangeGradDate"
326 13 Application.SetOption "Confirm Action Queries", True
327 14
328 15 ' Update the drop down list for MCRD Grad Date
329 16 Me.cmbMCRDGradDate.RowSource = "SELECT DISTINCTROW
330 17 (qryMCTGradDate) [GradDate] FROM (qryMCTGradDate);"
331 18
332 19 ' Force user to refresh data before performing another date change
333 20 Me.Refresh
334 21 Me.tblMCTGradDate.Enabled = False
335 22 Me.tblMCTGradDate.Enabled = False
336 23 Me.tblMCTGradDate.Enabled = False
337 24
338 25 ' Provide a message indicating the change was made
339 26 MsgBox "Date change was successful"
340 27
341 28 End Sub
342 29 Private Sub btnEquals_Click()
343 30 Dim varDate As Variant
344 31 Dim inDays As Integer
345 32
346 33 inDays = Me.cmbDays
347 34 varDate = Me.cmbMCRDGradDate
348 35
349 36 If IsNull(varDate) = False And IsNull(inDays) = False Then
350 37
351 38 ' Format this date for use in the DateAdd function
352 39 varDate = Format(varDate, "####/##/##")
353 40
354 41 ' Add 120 days to this date
355 42 varDate = DateAdd("d", inDays, varDate)
356 43
357 44 ' Change calendar to this date
358 45 Me.Calendar.Value = varDate
359 46
360 47 ' Convert the new date back into the original format
361 48 varDate = Format(varDate, "yyyy/mm/dd")
362 49
363 50 ' Place the new date in the "upper date bound" text box on the
364 51 calling form.
365 52 Me.tblMCTGradDate = varDate
366 53
367 54 ' Allows user to change the graduation date in the MARINE table
368 55 Me.tblMCTGradDate.Enabled = True
369 56 End If
370 57
371 58 End Sub
372 59 Private Sub cmbMCRDGradDate_AfterUpdate()
373 60 Dim varDate As Variant
374 61
375 62 varDate = Me.cmbMCRDGradDate
376 63 Me.tblMCTGradDate.Value = varDate
377 64
378 65 If IsNull(varDate) = False Then
379 66
380 67 ' Format this date for use in the DateAdd function
381 68 varDate = Format(varDate, "####/##/##")
382 69
383 70 ' Change calendar to this date
384 71 Me.Calendar.Value = varDate
385 72
386 73 Me.Calendar.Visible = True
387 74 Me.tblMCTGradDate.Enabled = True
388 75 Me.tblMCTGradDate.Enabled = True
389 76
390 77

```

```

391 78 Me.tblMCTGradDate.Enabled = True
392 79 End If
393 80 End Sub
394 81 Private Sub Form_Current()
395 82
396 83 Me.Calendar.Visible = False
397 84 Me.tblMCTGradDate.Enabled = False
398 85 Me.tblMCTGradDate.Enabled = False
399 86 Me.tblMCTGradDate.Enabled = False
400 87 Me.tblMCTGradDate.Enabled = False
401 88 Me.tblMCTGradDate.Enabled = False
402 89
403 90 End Sub
404 91 Private Sub Close_Click()
405 92 On Error GoTo Err_Close_Click
406 93 Dim stDocName As String
407 94 Dim stLinkCriteria As String
408 95
409 96 ' Close the current form
410 97 DoCmd.Close
411 98 ' Open specified form
412 99 stDocName = "frmImportExportSwitchboard"
413 100 DoCmd.OpenForm stDocName, , stLinkCriteria
414 101
415 102 Exit_L_Close_Click
416 103 Exit Sub
417 104
418 105 Err_Close_Click
419 106
420 107 MsgBox Err.Description
421 108 Resume Exit_L_Close_Click
422 109
423 110 End Sub
424 111
425 112
426 Form: frmClass
427 Code
428 1 Attribute VB_Name = "Form_frmClass"
429 2 Attribute VB_Creatable = True
430 3 Attribute VB_PredeclaredId = True
431 4 Attribute VB_Exposed = False
432 5 Option Compare Database
433 6 Option Explicit
434 7
435 8 Private Sub btnFindSchool_Click()
436 9 On Error GoTo Err_btnFindSchool_Click
437 10
438 11 Screen.PreviousControl.SetFocus
439 12 DoCmd.OpenForm acFormBar, acEditMenu, 10, , acMenuVer70
440 13
441 14 Exit_L_btnFindSchool_Click
442 15 Exit Sub
443 16
444 17 Err_btnFindSchool_Click
445 18 MsgBox Err.Description
446 19 Resume Exit_L_btnFindSchool_Click
447 20
448 21 End Sub
449 22
450 23
451 24 Form: frmClassQuotaPenaltyAndFit
452 Code
453 1 Attribute VB_Name = "Form_frmClassQuotaPenaltyAndFit"
454 2 Attribute VB_Creatable = True
455 3 Attribute VB_PredeclaredId = True
456 4 Attribute VB_Exposed = False
457 5 Option Compare Database
458 6 Option Explicit
459 7
460 8 Private Sub btnGo_Click()
461 9
462 10 Dim varUBound As Variant
463 11 Dim ingHour As Long, ingMin As Long, ingSec As Long
464 12 Dim ingStart As Long, ingEnd As Long
465 13 Dim strSQL As String
466 14
467 15 ' Starts the timer for displaying the time taken for this subroutine
468 16 ingStart = Timer
469 17
470 18 ' Prevents the user from going back to change the MCTGradDate until a
471 19 fitness file is
472 20 ' made, or the reset button is pushed.
473 21 Me.tblMCTGradDate.Enabled = False
474 22 varUBound = Me.UBound
475 23
476 24 ' Generate the quota and penalty files
477 25 Quota_Penalty (varUBound)
478 26
479 27 ' Update the quota/penalty records on this form
480 28 strSQL = "SELECT DISTINCTROW ([CLASS]) [CourseNumber_FK], " & _
481 29 "[CLASS] [AMOS_FK], [CLASS] [ClassIndex_FK], [CLASS] [ReportDate], " & _
482 30 "[CLASS] [Quota], [CLASS] [ClassIndex], [PENALTY] [ClassIndex_FK], " & _
483 31 "[PENALTY] [Penalty] FROM ([CLASS]) INNER JOIN ([PENALTY]) ON " & _
484 32 "[CLASS] [ClassIndex] = [PENALTY] [ClassIndex_FK];"
485 33 Me.RecordSource = strSQL
486 34
487 35 ' Update status of status text box
488 36 Me.Status = "Working..."
489 37
490 38 ' Call the Fitness procedure
491 39 Fitness
492 40
493 41 ' Update status of status text box
494 42 Me.Status = "Files generated"
495 43 DoCmd.Beep
496 44
497 45 ' Records the stop time for running this subroutine
498 46 ingEnd = Timer
499 47
500 48 ' Determine total secs
501 49 ingSec = ingEnd - ingStart
502 50
503 51 ' Determine total hours
504 52 ingHour = ingSec \ 3600
505 53
506 54 ' Determine total mins
507 55 ingMin = (ingSec - ingHour * 3600) \ 60
508 56
509 57 ' Determine the secs remaining
510 58 ingSec = (ingSec - (ingHour * 3600) - (ingMin * 60)) \ 1
511 59
512 60 MsgBox "This operation took " & ingHour & " hour(s), " & ingMin & "
513 61 min(s), " & _
514 62 "and " & ingSec & " secs.", vbOKOnly, "Recruit Distribution Model"
515 63
516 64 Me.tblMCTGradDate.Enabled = True
517 65 Me.tblMCTGradDate.Enabled = True
518 66 End Sub
519 67 Private Sub btnGo_MouseDown(Button As Integer, Shift As Integer, X As
520 68 Y As Single)
521 69
522 70

```

```

521 66 Me!Status = "Working..."
522 67
523 68 End Sub
524 69
525 70 Private Sub btnPrepareAndExecuteSolve_Click()
526 71 On Error GoTo Err_btnPrepareAndExecuteSolve_Click
527 72
528 73 Dim stDocName As String
529 74 Dim stLinkCriteria As String
530 75
531 76 ' Close the current form
532 77 DoCmd.Close
533 78 ' Open specified form
534 79 stDocName = "frmPrepareAndExecuteSolve"
535 80 DoCmd.OpenForm stDocName, , stLinkCriteria
536 81
537 82 Exit_btnPrepareAndExecuteSolve_Click
538 83 Exit Sub
539 84
540 85 Err_btnPrepareAndExecuteSolve_Click
541 86 MsgBox Err.Description
542 87 Resume Exit_btnPrepareAndExecuteSolve_Click
543 88
544 89 End Sub
545 90
546 91 Private Sub btnReturnPrevious_Click()
547 92 On Error GoTo Err_btnReturnPrevious_Click
548 93
549 94 Dim stDocName As String
550 95 Dim stLinkCriteria As String
551 96
552 97 ' Close the current form
553 98 DoCmd.Close
554 99 ' Open specified form
555 100 stDocName = "frmPreprocessingAndExecutionSwitchboard"
556 101 DoCmd.OpenForm stDocName, , stLinkCriteria
557 102
558 103 Exit_btnReturnPrevious_Click
559 104 Exit Sub
560 105
561 106 Err_btnReturnPrevious_Click
562 107 MsgBox Err.Description
563 108 Resume Exit_btnReturnPrevious_Click
564 109
565 110 End Sub
566 111
567 112 Private Sub cmbCourseNumberFind_AfterUpdate()
568 113
569 114 Dim R As Recordset
570 115 Set R = Me.RecordsetClone
571 116 R.FindFirst "SCourseNumber_PK = " & Chr(34) &
572 117 Me!cmbCourseNumberFind & Chr(34)
573 118 Me!cmbCourseNumberFind = Null
574 119 Me.Penalty.SetFocus
575 120
576 121 End Sub
577 122
578 123 Private Sub Form_Current()
579 124
580 125 ' Prevents the user from creating the AMPL files until a MCT
581 126 ' graduation date is entered
582 127 If IsNull(Me.MCTGradDate.Value) Then
583 128 Me!btnGo.Enabled = False
584 129 Me!Status.Enabled = False
585 130
586 131 ' Allow the user to create the quote and penalty files
587 132 Else
588 133 Me!btnGo.Enabled = True
589 134
590 135 End If
591 136 ' Update the status of the status text box
592 137 Me!Status = ""
593 138
594 139 End Sub
595 140
596 141 Private Sub MCTGradDate_AfterUpdate()
597 142
598 143 Dim varDate As Variant
599 144
600 145 varDate = Me.MCTGradDate
601 146
602 147 If IsNull(varDate) = False Then
603 148
604 149 Me!btnGo.Enabled = True
605 150 Me!Status.Enabled = True
606 151
607 152 ' Format this date for use in the DateAdd function
608 153 varDate = Format(varDate, "####/##/##")
609 154
610 155 ' Add 120 days to this date
611 156 varDate = DateAdd("d", 120, varDate)
612 157
613 158 ' Convert the new date back into the original format
614 159 varDate = Format(varDate, "yyyymmdd")
615 160
616 161 ' Place the new date in the "upper date bound" text box on the
617 162 ' calling form.
618 163 Me!UpperDateBound = varDate
619 164
620 165 Else
621 166 Me!btnGo.Enabled = False
622 167 Me!Status.Enabled = False
623 168 End If
624 169
625 170 End Sub
626 171
627 172
628 173
629 174
630 175
631 176
632 177
633 178
634 179
635 180
636 181
637 182
638 183
639 184
640 185
641 186
642 187
643 188
644 189
645 190
646 191
647 192
648 193
649 194
650 195

```

```

651 21 Resume Exit_btnFind_Click
652 22 End Sub
653 23
654 24
655 25
656 26 Form: frmFundamentalProperty
657 27
658 28 Attribute VB_Name = "Form_frmFundamentalProperty"
659 29
660 30 Attribute VB_Creatable = True
661 31 Attribute VB_PredeclaredId = True
662 32 Attribute VB_Exposed = False
663 33
664 34 Option Compare Database
665 35 Option Explicit
666 36
667 37 Private Sub Command15_Click()
668 38
669 39 MsgBox "This is a button", vbInformation, "Testing"
670 40
671 41 End Sub
672 42
673 43
674 44
675 45
676 46
677 47
678 48
679 49
680 50
681 51
682 52
683 53
684 54
685 55
686 56
687 57
688 58
689 59
690 60
691 61
692 62
693 63
694 64
695 65
696 66
697 67
698 68
699 69
700 70
701 71
702 72
703 73
704 74
705 75
706 76
707 77
708 78
709 79
710 80
711 81
712 82
713 83
714 84
715 85
716 86
717 87
718 88
719 89
720 90
721 91
722 92
723 93
724 94
725 95
726 96
727 97
728 98
729 99
730 100
731 101
732 102
733 103
734 104
735 105
736 106
737 107
738 108
739 109
740 110
741 111
742 112
743 113
744 114
745 115
746 116
747 117
748 118
749 119
750 120
751 121
752 122
753 123
754 124
755 125
756 126
757 127
758 128
759 129
760 130
761 131
762 132
763 133
764 134
765 135
766 136
767 137
768 138
769 139
770 140
771 141
772 142
773 143
774 144
775 145
776 146
777 147
778 148
779 149
780 150

```

```

781 117 ' Forces user to enter data for property name and description
782 118 If IsNull(Me.FPropertyName_PK.Value) Or IsNull(Me.Description.Value)
783 119 Me.MarineField.Enabled = False
784 120 Me.Operator.Enabled = False
785 121 Me.LabID.Visible = False
786 122 Me.SubformFundamentalProperty.Visible = False
787 123 Me.SubformFundamentalPropertyList.Visible = False
788 124
789 125 ' Allows the user to enter values for the fundamental equation
790 126 Else
791 127 Me.MarineField.Enabled = True
792 128 Me.Operator.Enabled = True
793 129 Me.LabID.Visible = True
794 130 If Form!frmFundamentalProperty.Operator.Value = "in" Then
795 131 ' Update the Potential Value list
796 132 Me.SubformFundamentalProperty.Visible = False
797 133 Me.SubformFundamentalPropertyList.Visible = True
798 134 strSQL = "SELECT DISTINCT * & Me.MarineField & " FROM
799 135
800 Form!frmFundamentalProperty!SubformFundamentalPropertyList!Value_FK.RowSo
801 urce = strSQL
802 136 Else
803 137 ' Update the Potential Value list
804 138 strSQL = "SELECT DISTINCT * & Me.MarineField & " FROM
805 139
806 Form!frmFundamentalProperty!SubformFundamentalProperty!cmbValue_FK.RowSou
807 rce = strSQL
808 140 Me.SubformFundamentalProperty.Visible = True
809 141 Me.SubformFundamentalPropertyList.Visible = False
810 142 End If
811 143
812 144 ' Set focus to the property name. This prevents problem
813 associated with entering the "Operator" object.
814 145 FPropertyName_PK.SetFocus
815 146 End If
816 147
817 148
818 149
819 150 End Sub
820 151
821 152 Private Sub FPropertyName_PK_AfterUpdate()
822 153
823 154 Dim strSQL As String
824 155
825 156 ' Forces user to enter data for property name and description
826 157 If IsNull(Me.FPropertyName_PK.Value) Or IsNull(Me.Description.Value)
827 158 Me.MarineField.Enabled = False
828 159 Me.Operator.Enabled = False
829 160 Me.LabID.Visible = False
830 161 Me.SubformFundamentalProperty.Visible = False
831 162 Me.SubformFundamentalPropertyList.Visible = False
832 163
833 164 ' Allows the user to enter values for the fundamental equation
834 165 Else
835 166 Me.MarineField.Enabled = True
836 167 Me.Operator.Enabled = True
837 168 Me.LabID.Visible = True
838 169 If Form!frmFundamentalProperty.Operator.Value = "in" Then
839 170 strSQL = "SELECT DISTINCT * & Me.MarineField & " FROM
840 171
841 Form!frmFundamentalProperty!SubformFundamentalPropertyList!Value_FK.RowSo
842 urce = strSQL
843 172 Me.SubformFundamentalProperty.Visible = False
844 173 Me.SubformFundamentalPropertyList.Visible = True
845 174
846 175 Else
847 176 strSQL = "SELECT DISTINCT * & Me.MarineField & " FROM
848 177
849 Form!frmFundamentalProperty!SubformFundamentalProperty!cmbValue_FK.RowSou
850 rce = strSQL
851 178 Me.SubformFundamentalProperty.Visible = True
852 179 Me.SubformFundamentalPropertyList.Visible = False
853 180 End If
854 181 End If
855 182
856 183 End Sub
857 184
858 185 Private Sub MarineField_AfterUpdate()
859 186
860 187 Dim strSQL As String
861 188
862 189 ' Update the Potential Value list
863 190 strSQL = "SELECT DISTINCT * & Me.MarineField & " FROM MARINE:"
864 191
865 Form!frmFundamentalProperty!SubformFundamentalPropertyList!Value_FK.RowSo
866 urce = strSQL
867 192
868 Form!frmFundamentalProperty!SubformFundamentalProperty!cmbValue_FK.RowSou
869 rce = strSQL
870 193
871 194 End Sub
872 195
873 196 Private Sub Operator_AfterUpdate()
874 197
875 198 If Form!frmFundamentalProperty.Operator.Value = "in" Then
876 199 Me.SubformFundamentalProperty.Visible = False
877 200 Me.SubformFundamentalPropertyList.Visible = True
878 201 Else
879 202 Me.SubformFundamentalProperty.Visible = True
880 203 Me.SubformFundamentalPropertyList.Visible = False
881 204 End If
882 205
883 206 End Sub
884 207 Private Sub btnDelete_Click()
885 208 On Error GoTo Err_btnDelete_Click
886 209
887 210
888 211 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
889 212 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
890 213
891 214 Exit btnDelete_Click
892 215 Exit Sub
893 216
894 217 Err_btnDelete_Click
895 218 MsgBox Err.Description
896 219 Resume Exit_btnDelete_Click
897 220
898 221 End Sub
899 222 Private Sub Command14_Click()
900 223 On Error GoTo Err_Command14_Click
901 224
902 225 Dim sDialStr As String
903 226 Dim PrevCtrl As Control
904 227 Const ERR_OBNOTEXIST = 2467
905 228 Const ERR_OBNOTSET = 91
906 229
907 230 Set PrevCtrl = Screen.PreviousControl
908 231
909 232 If TypeOf PrevCtrl Is TextBox Then
910 233 sDialStr = IIf(VarType(PrevCtrl) > V_NULL, PrevCtrl, "")

```

```

911 234 ElseIf TypeOf PrevCtrl Is ListBox Then
912 235 sDialStr = IIf(VarType(PrevCtrl) > V_NULL, PrevCtrl, "")
913 236 ElseIf TypeOf PrevCtrl Is ComboBox Then
914 237 sDialStr = IIf(VarType(PrevCtrl) > V_NULL, PrevCtrl, "")
915 238 Else
916 239 sDialStr = ""
917 240 End If
918 241
919 242 Application.Run "Utility.wbt_AutoDial", sDialStr
920 243
921 244 Exit_Command14_Click
922 245 Exit Sub
923 246
924 247 Err_Command14_Click
925 248 If (Err = ERR_OBNOTEXIST) Or (Err = ERR_OBNOTSET) Then
926 249 Resume Next
927 250 End If
928 251 MsgBox Err.Description
929 252 Resume Exit_Command14_Click
930 253
931 254 End Sub
932 255
933 256 Private Sub Operator_Enter()
934 257
935 258 If Me.Operator.Value = "in" Then
936 259 MsgBox "The 'in' operator is not changeable. It is removed by
937 deleting the property.", vbInformation, "Unauthorized Action"
938 260 Me.SubformFundamentalPropertyList.SetFocus
939 261 End If
940 262
941 263 End Sub
942 264
943 Form: frmGenerateReport
944 Code
945 1 Attribute VB_Name = "Form_frmGenerateReport"
946 2 Attribute VB_Creatable = True
947 3 Attribute VB_PredeclaredId = True
948 4 Attribute VB_Exposed = False
949 5 Option Compare Database
950 6 Option Explicit
951 7
952 8 Private Sub btnAssignmentReport_Click()
953 9
954 10 DoCmd.OpenReport "rptAssignedMarines", acViewPreview
955 11
956 12 End Sub
957 13
958 14 Private Sub btnEntryLevelSchool_Click()
959 15
960 16 DoCmd.OpenReport "rptEntryLevelSchool", acViewPreview
961 17
962 18 End Sub
963 19 End Sub
964 20
965 21 Private Sub btnExit_Click()
966 22 On Error GoTo Err_btnExit_Click
967 23
968 24 DoCmd.Close
969 25
970 26
971 27 Exit_btnExit_Click
972 28 Exit Sub
973 29
974 30 Err_btnExit_Click
975 31 MsgBox Err.Description
976 32 Resume Exit_btnExit_Click
977 33
978 34
979 35 End Sub
980 36
981 37 Private Sub btnUnassignedMarineReport_Click()
982 38
983 39 DoCmd.OpenReport "rptUnassignedMarines", acViewPreview
984 40
985 41 End Sub
986 42
987 43 Private Sub Command8_Click()
988 44 On Error GoTo Err_Command8_Click
989 45
990 46 Dim sDocName As String
991 47 Dim sLinkCriteria As String
992 48
993 49 ' Close the current form
994 50 DoCmd.Close
995 51 ' Open specified form
996 52 sDocName = "frmRDM_Main_Switchboard"
997 53 DoCmd.OpenForm sDocName, , sLinkCriteria
998 54
999 55 Exit_Command8_Click
1000 56 Exit Sub
1001 57
1002 58 Err_Command8_Click
1003 59 MsgBox Err.Description
1004 60 Resume Exit_Command8_Click
1005 61
1006 62 End Sub
1007
1008 Form: frmImport/ExportSwitchboard
1009 Code
1010 1 Attribute VB_Name = "Form_frmImport/ExportSwitchboard"
1011 2 Attribute VB_Creatable = True
1012 3 Attribute VB_PredeclaredId = True
1013 4 Attribute VB_Exposed = False
1014 5 Option Compare Database
1015 6 Option Explicit
1016 7
1017 8 Private Sub btnExit_Click()
1018 9 On Error GoTo Err_btnExit_Click
1019 10
1020 11 DoCmd.Close
1021 12
1022 13 Exit_btnExit_Click
1023 14 Exit Sub
1024 15
1025 16
1026 17 Err_btnExit_Click
1027 18 MsgBox Err.Description
1028 19 Resume Exit_btnExit_Click
1029 20
1030 21
1031 22 End Sub
1032 23
1033 24 Private Sub btnExportR3_Click()
1034 25
1035 26 Application.SetOption "Confirm Action Queries", False
1036 27 DoCmd.SaveWarnings (False)
1037 28 ' Exports assignments to R33.bst file
1038 29 ExportR33
1039 30 ' Archives data from the Marine and assignment tables, having report
1040 31 dates less than the current date

```

```

1041 31 ArchiveData
1042 32 Application.SetOption "Confirm Action Queries", True
1043 33 DoCmd.SetWarnings (True)
1044 34
1045 35 End Sub
1046 36
1047 37
1048 38 Private Sub Command2_Click()
1049 39
1050 40 Dim stDocName As String
1051 41 Dim stLinkCriteria As String
1052 42
1053 43 Application.SetOption "Confirm Action Queries", False
1054 44 DoCmd.SetWarnings (False)
1055 45 ' Imports the RDI file
1056 46 Import RDI
1057 47 Application.SetOption "Confirm Action Queries", True
1058 48 DoCmd.SetWarnings (True)
1059 49
1060 50 ' Close the current form
1061 51 DoCmd.Close
1062 52 ' Open specified form
1063 53 stDocName = "frmChangeGradDate"
1064 54 DoCmd.OpenForm stDocName, , , stLinkCriteria
1065 55
1066 56 End Sub
1067 57
1068 58 Private Sub Command4_Click()
1069 59
1070 60 ' Imports BNA file
1071 61 Application.SetOption "Confirm Action Queries", False
1072 62 DoCmd.SetWarnings (False)
1073 63 DoCmd.RunSQL "DELETE FROM BNA_EXTRACT;"
1074 64 Import BNA
1075 65 Application.SetOption "Confirm Action Queries", True
1076 66 DoCmd.SetWarnings (True)
1077 67
1078 68 End Sub
1079 69
1080 70 Private Sub Command8_Click()
1081 71
1082 72 On Error GoTo Err_Command8_Click
1083 73
1084 74 Dim stDocName As String
1085 75 Dim stLinkCriteria As String
1086 76
1087 77 ' Close the current form
1088 78 DoCmd.Close
1089 79 ' Open specified form
1090 80 stDocName = "frmRDM_Main_Switchboard"
1091 81 DoCmd.OpenForm stDocName, , , stLinkCriteria
1092 82
1093 83 Exit_Command8_Click
1094 84 Exit Sub
1095 85
1096 86 Err_Command8_Click
1097 87 MsgBox Err.Description
1098 88 Resume Exit_Command8_Click
1099 89
1100 90 End Sub
1101
1102 Form: frmLogicalProperty
1103 Code
1104 1 Attribute VB_Name = "Form_frmLogicalProperty"
1105 2 Attribute VB_Creatable = True
1106 3 Attribute VB_PredeclaredId = True
1107 4 Attribute VB_Exposed = False
1108 5 Option Compare Database
1109 6 Option Explicit
1110 7
1111 8 Private Sub btnClose_Click()
1112 9 On Error GoTo Err_btnClose_Click
1113 10
1114 11 DoCmd.Close
1115 12 DoCmd.OpenForm "frmMaintenanceSwitchboard"
1116 13
1117 14 Exit_btnClose_Click
1118 15 Exit Sub
1119 16
1120 17 Err_btnClose_Click
1121 18 MsgBox Err.Description
1122 19 Resume Exit_btnClose_Click
1123 20
1124 21 End Sub
1125 22
1126 23 Private Sub cmbFundProp_Click()
1127 24
1128 25 ' Adds the selected fundamental property to the logical equation
1129 26 Me.LogicalEquation.Value = Me.LogicalEquation.Value &
1130 27 Me.cmbFundProp.Value & " "
1131 28
1132 29 End Sub
1133 30 Private Sub cmbLogOper_Click()
1134 31
1135 32 ' Adds the selected logical operator to the logical equation
1136 33 Me.LogicalEquation.Value = Me.LogicalEquation.Value &
1137 34 Me.cmbLogOper.Value & " "
1138 35 End Sub
1139 36
1140 37 Private Sub cmbPropertyNameFind_AfterUpdate()
1141 38
1142 39 Dim R As Recordset
1143 40 Set R = Me.RecordsetClone
1144 41 R.FindFirst "[!PropertyName_PK] = " & Chr(34) &
1145 42 Me.cmbPropertyNameFind & Chr(34)
1146 43 Me.Bookmark = R.Bookmark
1147 44 Me.cmbPropertyNameFind = Null
1148 45 Me.LPPropertyName_PK.SetFocus
1149 46 End Sub
1150 47
1151 48 Private Sub Command10_Click()
1152 49
1153 50 ' Removes the last entry in the logical equation
1154 51 Dim stInName, strUndo As String ' input name
1155 52 Dim lnSpacePos As Integer ' position of the space
1156 53 Dim l As Integer ' loop index
1157 54 Dim boofResult As Boolean
1158 55 Dim lngConversion As Long
1159 56 Dim varConv As Variant
1160 57
1161 58 ' get the logical expression
1162 59 strInName = Me.LogicalEquation.Value
1163 60
1164 61 ' Check for "Null" value
1165 62 If strInName <> "Null" Then
1166 63
1167 64
1168 65
1169 66
1170 67
1171 68
1172 69
1173 70
1174 71
1175 72
1176 73
1177 74
1178 75
1179 76
1180 77
1181 78
1182 79
1183 80
1184 81
1185 82
1186 83
1187 84
1188 85
1189 86
1190 87
1191 88
1192 89
1193 90
1194 91
1195 92
1196 93
1197 94
1198 95
1199 96
1200 97
1201 98
1202 99
1203 100
1204 101
1205 102
1206 103
1207 104
1208 105
1209 106
1210 107
1211 108
1212 109
1213 110
1214 111
1215 112
1216 113
1217 114
1218 115
1219 116
1220 117
1221 118
1222 119
1223 120
1224 121
1225 122
1226 123
1227 124
1228 125
1229 126
1230 127
1231 128
1232 129
1233 130
1234 131
1235 132
1236 133
1237 134
1238 135
1239 136
1240 137
1241 138
1242 139
1243 140
1244 141
1245 142
1246 143
1247 144
1248 145
1249 146
1250 147
1251 148
1252 149
1253 150
1254 151
1255 152
1256 153
1257 154
1258 155
1259 156
1260 157
1261 158
1262 159
1263 160
1264 161
1265 162
1266 163
1267 164
1268 165
1269 166
1270 167
1271 168
1272 169
1273 170
1274 171
1275 172
1276 173
1277 174
1278 175
1279 176
1280 177
1281 178
1282 179
1283 180
1284 181
1285 182
1286 183
1287 184
1288 185
1289 186
1290 187
1291 188
1292 189
1293 190
1294 191
1295 192
1296 193
1297 194
1298 195
1299 196
1300 197
1301 198
1302 199
1303 200
1304 201
1305 202
1306 203
1307 204
1308 205
1309 206
1310 207
1311 208
1312 209
1313 210
1314 211
1315 212
1316 213
1317 214
1318 215
1319 216
1320 217
1321 218
1322 219
1323 220
1324 221
1325 222
1326 223
1327 224
1328 225
1329 226
1330 227
1331 228
1332 229
1333 230
1334 231
1335 232
1336 233
1337 234
1338 235
1339 236
1340 237
1341 238
1342 239
1343 240
1344 241
1345 242
1346 243
1347 244
1348 245
1349 246
1350 247
1351 248
1352 249
1353 250
1354 251
1355 252
1356 253
1357 254
1358 255
1359 256
1360 257
1361 258
1362 259
1363 260
1364 261
1365 262
1366 263
1367 264
1368 265
1369 266
1370 267
1371 268
1372 269
1373 270
1374 271
1375 272
1376 273
1377 274
1378 275
1379 276
1380 277
1381 278
1382 279
1383 280
1384 281
1385 282
1386 283
1387 284
1388 285
1389 286
1390 287
1391 288
1392 289
1393 290
1394 291
1395 292
1396 293
1397 294
1398 295
1399 296
1400 297
1401 298
1402 299
1403 300
1404 301
1405 302
1406 303
1407 304
1408 305
1409 306
1410 307
1411 308
1412 309
1413 310
1414 311
1415 312
1416 313
1417 314
1418 315
1419 316
1420 317
1421 318
1422 319
1423 320
1424 321
1425 322
1426 323
1427 324
1428 325
1429 326
1430 327
1431 328
1432 329
1433 330
1434 331
1435 332
1436 333
1437 334
1438 335
1439 336
1440 337
1441 338
1442 339
1443 340
1444 341
1445 342
1446 343
1447 344
1448 345
1449 346
1450 347
1451 348
1452 349
1453 350
1454 351
1455 352
1456 353
1457 354
1458 355
1459 356
1460 357
1461 358
1462 359
1463 360
1464 361
1465 362
1466 363
1467 364
1468 365
1469 366
1470 367
1471 368
1472 369
1473 370
1474 371
1475 372
1476 373
1477 374
1478 375
1479 376
1480 377
1481 378
1482 379
1483 380
1484 381
1485 382
1486 383
1487 384
1488 385
1489 386
1490 387
1491 388
1492 389
1493 390
1494 391
1495 392
1496 393
1497 394
1498 395
1499 396
1500 397
1501 398
1502 399
1503 400
1504 401
1505 402
1506 403
1507 404
1508 405
1509 406
1510 407
1511 408
1512 409
1513 410
1514 411
1515 412
1516 413
1517 414
1518 415
1519 416
1520 417
1521 418
1522 419
1523 420
1524 421
1525 422
1526 423
1527 424
1528 425
1529 426
1530 427
1531 428
1532 429
1533 430
1534 431
1535 432
1536 433
1537 434
1538 435
1539 436
1540 437
1541 438
1542 439
1543 440
1544 441
1545 442
1546 443
1547 444
1548 445
1549 446
1550 447
1551 448
1552 449
1553 450
1554 451
1555 452
1556 453
1557 454
1558 455
1559 456
1560 457
1561 458
1562 459
1563 460
1564 461
1565 462
1566 463
1567 464
1568 465
1569 466
1570 467
1571 468
1572 469
1573 470
1574 471
1575 472
1576 473
1577 474
1578 475
1579 476
1580 477
1581 478
1582 479
1583 480
1584 481
1585 482
1586 483
1587 484
1588 485
1589 486
1590 487
1591 488
1592 489
1593 490
1594 491
1595 492
1596 493
1597 494
1598 495
1599 496
1600 497
1601 498
1602 499
1603 500
1604 501
1605 502
1606 503
1607 504
1608 505
1609 506
1610 507
1611 508
1612 509
1613 510
1614 511
1615 512
1616 513
1617 514
1618 515
1619 516
1620 517
1621 518
1622 519
1623 520
1624 521
1625 522
1626 523
1627 524
1628 525
1629 526
1630 527
1631 528
1632 529
1633 530
1634 531
1635 532
1636 533
1637 534
1638 535
1639 536
1640 537
1641 538
1642 539
1643 540
1644 541
1645 542
1646 543
1647 544
1648 545
1649 546
1650 547
1651 548
1652 549
1653 550
1654 551
1655 552
1656 553
1657 554
1658 555
1659 556
1660 557
1661 558
1662 559
1663 560
1664 561
1665 562
1666 563
1667 564
1668 565
1669 566
1670 567
1671 568
1672 569
1673 570
1674 571
1675 572
1676 573
1677 574
1678 575
1679 576
1680 577
1681 578
1682 579
1683 580
1684 581
1685 582
1686 583
1687 584
1688 585
1689 586
1690 587
1691 588
1692 589
1693 590
1694 591
1695 592
1696 593
1697 594
1698 595
1699 596
1700 597
1701 598
1702 599
1703 600
1704 601
1705 602
1706 603
1707 604
1708 605
1709 606
1710 607
1711 608
1712 609
1713 610
1714 611
1715 612
1716 613
1717 614
1718 615
1719 616
1720 617
1721 618
1722 619
1723 620
1724 621
1725 622
1726 623
1727 624
1728 625
1729 626
1730 627
1731 628
1732 629
1733 630
1734 631
1735 632
1736 633
1737 634
1738 635
1739 636
1740 637
1741 638
1742 639
1743 640
1744 641
1745 642
1746 643
1747 644
1748 645
1749 646
1750 647
1751 648
1752 649
1753 650
1754 651
1755 652
1756 653
1757 654
1758 655
1759 656
1760 657
1761 658
1762 659
1763 660
1764 661
1765 662
1766 663
1767 664
1768 665
1769 666
1770 667
1771 668
1772 669
1773 670
1774 671
1775 672
1776 673
1777 674
1778 675
1779 676
1780 677
1781 678
1782 679
1783 680
1784 681
1785 682
1786 683
1787 684
1788 685
1789 686
1790 687
1791 688
1792 689
1793 690
1794 691
1795 692
1796 693
1797 694
1798 695
1799 696
1800 697
1801 698
1802 699
1803 700
1804 701
1805 702
1806 703
1807 704
1808 705
1809 706
1810 707
1811 708
1812 709
1813 710
1814 711
1815 712
1816 713
1817 714
1818 715
1819 716
1820 717
1821 718
1822 719
1823 720
1824 721
1825 722
1826 723
1827 724
1828 725
1829 726
1830 727
1831 728
1832 729
1833 730
1834 731
1835 732
1836 733
1837 734
1838 735
1839 736
1840 737
1841 738
1842 739
1843 740
1844 741
1845 742
1846 743
1847 744
1848 745
1849 746
1850 747
1851 748
1852 749
1853 750
1854 751
1855 752
1856 753
1857 754
1858 755
1859 756
1860 757
1861 758
1862 759
1863 760
1864 761
1865 762
1866 763
1867 764
1868 765
1869 766
1870 767
1871 768
1872 769
1873 770
1874 771
1875 772
1876 773
1877 774
1878 775
1879 776
1880 777
1881 778
1882 779
1883 780
1884 781
1885 782
1886 783
1887 784
1888 785
1889 786
1890 787
1891 788
1892 789
1893 790
1894 791
1895 792
1896 793
1897 794
1898 795
1899 796
1900 797
1901 798
1902 799
1903 800
1904 801
1905 802
1906 803
1907 804
1908 805
1909 806
1910 807
1911 808
1912 809
1913 810
1914 811
1915 812
1916 813
1917 814
1918 815
1919 816
1920 817
1921 818
1922 819
1923 820
1924 821
1925 822
1926 823
1927 824
1928 825
1929 826
1930 827
1931 828
1932 829
1933 830
1934 831
1935 832
1936 833
1937 834
1938 835
1939 836
1940 837
1941 838
1942 839
1943 840
1944 841
1945 842
1946 843
1947 844
1948 845
1949 846
1950 847
1951 848
1952 849
1953 850
1954 851
1955 852
1956 853
1957 854
1958 855
1959 856
1960 857
1961 858
1962 859
1963 860
1964 861
1965 862
1966 863
1967 864
1968 865
1969 866
1970 867
1971 868
1972 869
1973 870
1974 871
1975 872
1976 873
1977 874
1978 875
1979 876
1980 877
1981 878
1982 879
1983 880
1984 881
1985 882
1986 883
1987 884
1988 885
1989 886
1990 887
1991 888
1992 889
1993 890
1994 891
1995 892
1996 893
1997 894
1998 895
1999 896
2000 897
2001 898
2002 899
2003 900
2004 901
2005 902
2006 903
2007 904
2008 905
2009 906
2010 907
2011 908
2012 909
2013 910
2014 911
2015 912
2016 913
2017 914
2018 915
2019 916
2020 917
2021 918
2022 919
2023 920
2024 921
2025 922
2026 923
2027 924
2028 925
2029 926
2030 927
2031 928
2032 929
2033 930
2034 931
2035 932
2036 933
2037 934
2038 935
2039 936
2040 937
2041 938
2042 939
2043 940
2044 941
2045 942
2046 943
2047 944
2048 945
2049 946
2050 947
2051 948
2052 949
2053 950
2054 951
2055 952
2056 953
2057 954
2058 955
2059 956
2060 957
2061 958
2062 959
2063 960
2064 961
2065 962
2066 963
2067 964
2068 965
2069 966
2070 967
2071 968
2072 969
2073 970
2074 971
2075 972
2076 973
2077 974
2078 975
2079 976
2080 977
2081 978
2082 979
2083 980
2084 981
2085 982
2086 983
2087 984
2088 985
2089 986
2090 987
2091 988
2092 989
2093 990
2094 991
2095 992
2096 993
2097 994
2098 995
2099 996
2100 997
2101 998
2102 999
2103 1000

```

```

1301 188 Me!Command8.Enabled = True
1302 189 Me!Command9.Enabled = True
1303 190 Me!Command10.Enabled = True
1304 191 Me!cmbFundProp.Enabled = True
1305 192 Me!cmbLogOper.Enabled = True
1306 193 Me!LogicalEquation.Enabled = True
1307 194 Me!LogicalEquation.Value = "("
1308 195
1309 196 ' Allows the user to enter values into the logical equation
1310 197 Else
1311 198 Me!Command8.Enabled = True
1312 199 Me!Command9.Enabled = True
1313 200 Me!Command10.Enabled = True
1314 201 Me!cmbFundProp.Enabled = True
1315 202 Me!cmbLogOper.Enabled = True
1316 203 Me!LogicalEquation.Enabled = True
1317 204 End If
1318 205
1319 206 End Sub
1320 207
1321 208 Private Sub Form_BeforeUpdate(Cancel As Integer)
1322 209
1323 210 (LogPropTimeStamp) = Now
1324 211
1325 212 End Sub
1326 213
1327 214 Private Sub Form_Current()
1328 215
1329 216 ' Forces user to enter data for property name and description
1330 217 If IsNull(Me.LPropertyName_PK.Value) Or IsNull(Me.Description.Value)
1331 218 Me!Command8.Enabled = False
1332 219 Me!Command9.Enabled = False
1333 220 Me!Command10.Enabled = False
1334 221 Me!cmbFundProp.Enabled = False
1335 222 Me!cmbLogOper.Enabled = False
1336 223 Me!LogicalEquation.Enabled = False
1337 224
1338 225 ' Enters a "(" be the first entry in the logical equation if it is
1339 226 ElseIf IsNull(Me!LogicalEquation.Value) Then
1340 227 Me!LogicalEquation.Value = "("
1341 228
1342 229 ' Allows the user to enter values into the logical equation
1343 230 Else
1344 231 Me!Command8.Enabled = True
1345 232 Me!Command9.Enabled = True
1346 233 Me!Command10.Enabled = True
1347 234 Me!cmbFundProp.Enabled = True
1348 235 Me!cmbLogOper.Enabled = True
1349 236 Me!LogicalEquation.Enabled = True
1350 237 End If
1351 238
1352 239 End Sub
1353 240 Private Sub LogicalEquation_Enter()
1354 241
1355 242 ' Prevents the user from modifying the logical equation directly
1356 243 MsgBox "The logical equation is only modifiable by the above logical
1357 244 equation builder"
1358 245 DoCmd.GoToControl "LPropertyName_PK"
1359 246
1360 246 End Sub
1361 247
1362 248 Private Sub LPropertyName_PK_AfterUpdate()
1363 249
1364 250 ' Forces user to enter data for property name and description
1365 251 If IsNull(Me.LPropertyName_PK.Value) Or IsNull(Me.Description.Value)
1366 252 Me!Command8.Enabled = False
1367 253 Me!Command9.Enabled = False
1368 254 Me!Command10.Enabled = False
1369 255 Me!cmbFundProp.Enabled = False
1370 256 Me!cmbLogOper.Enabled = False
1371 257 Me!LogicalEquation.Enabled = False
1372 258
1373 259 ' Enters a "(" be the first entry in the logical equation if it is
1374 260 ElseIf IsNull(Me!LogicalEquation.Value) Then
1375 261 Me!Command8.Enabled = True
1376 262 Me!Command9.Enabled = True
1377 263 Me!Command10.Enabled = True
1378 264 Me!cmbFundProp.Enabled = True
1379 265 Me!cmbLogOper.Enabled = True
1380 266 Me!LogicalEquation.Enabled = True
1381 267 Me!LogicalEquation.Value = "("
1382 268
1383 269 ' Allows the user to enter values into the logical equation
1384 270 Else
1385 271 Me!Command8.Enabled = True
1386 272 Me!Command9.Enabled = True
1387 273 Me!Command10.Enabled = True
1388 274 Me!cmbFundProp.Enabled = True
1389 275 Me!cmbLogOper.Enabled = True
1390 276 Me!LogicalEquation.Enabled = True
1391 277 End If
1392 278
1393 279 End Sub
1394
1395 Form: frmMaintenanceSwitchboard
1396 Code
1397 1 Attribute VB_Name = "Form_frmMaintenanceSwitchboard"
1398 2 Attribute VB_Creatable = True
1399 3 Attribute VB_PredeclaredId = True
1400 4 Attribute VB_Exposed = False
1401 5 Option Compare Database
1402 6 Option Explicit
1403 7
1404 8 Private Sub btnExit_Click()
1405 9 On Error GoTo Err_btnExit_Click
1406 10
1407 11
1408 12 DoCmd.Close
1409 13
1410 14 Exit_btnExit_Click
1411 15 Exit Sub
1412 16
1413 17 Err_btnExit_Click
1414 18 MsgBox Err.Description
1415 19 Resume Exit_btnExit_Click
1416 20
1417 21
1418 22 End Sub
1419 23
1420 24 Private Sub Command0_Click()
1421 25 On Error GoTo Err_Command0_Click
1422 26
1423 27 Dim stDocName As String
1424 28 Dim stLinkCriteria As String
1425 29
1426 30 ' Close the current form
1427 31 DoCmd.Close
1428 32 ' Open specified form
1429 33 stDocName = "frmSchoolToPEF_Maint"
1430 34 DoCmd.OpenForm stDocName, , , stLinkCriteria

```

```

1431 35
1432 36 Exit_Command0_Click
1433 37 Exit Sub
1434 38
1435 39 Err_Command0_Click
1436 40 MsgBox Err.Description
1437 41 Resume Exit_Command0_Click
1438 42
1439 43 End Sub
1440 44
1441 45 Private Sub Command12_Click()
1442 46 On Error GoTo Err_Command12_Click
1443 47
1444 48 Dim stDocName As String
1445 49 Dim stLinkCriteria As String
1446 50
1447 51 ' Close the current form
1448 52 DoCmd.Close
1449 53 ' Open specified form
1450 54 stDocName = "frmSchools"
1451 55 DoCmd.OpenForm stDocName, , , stLinkCriteria
1452 56
1453 57 Exit_Command12_Click
1454 58 Exit Sub
1455 59
1456 60 Err_Command12_Click
1457 61 MsgBox Err.Description
1458 62 Resume Exit_Command12_Click
1459 63
1460 64 End Sub
1461 65
1462 66 Private Sub Command14_Click()
1463 67 On Error GoTo Err_Command14_Click
1464 68
1465 69 Dim stDocName As String
1466 70 Dim stLinkCriteria As String
1467 71
1468 72 ' Close the current form
1469 73 DoCmd.Close
1470 74 ' Open specified form
1471 75 stDocName = "frmPEF"
1472 76 DoCmd.OpenForm stDocName, , , stLinkCriteria
1473 77
1474 78 Exit_Command14_Click
1475 79 Exit Sub
1476 80
1477 81 Err_Command14_Click
1478 82 MsgBox Err.Description
1479 83 Resume Exit_Command14_Click
1480 84
1481 85 End Sub
1482 86
1483 87 Private Sub Command2_Click()
1484 88 On Error GoTo Err_Command2_Click
1485 89
1486 90 Dim stDocName As String
1487 91 Dim stLinkCriteria As String
1488 92
1489 93 ' Close the current form
1490 94 DoCmd.Close
1491 95 ' Open specified form
1492 96 stDocName = "frmFundamentalProperty"
1493 97 DoCmd.OpenForm stDocName, , , stLinkCriteria
1494 98
1495 99 Exit_Command2_Click
1496 100 Exit Sub
1497 101
1498 102 Err_Command2_Click
1499 103 MsgBox Err.Description
1500 104 Resume Exit_Command2_Click
1501 105
1502 106 End Sub
1503 107
1504 108 Private Sub Command4_Click()
1505 109 On Error GoTo Err_Command4_Click
1506 110
1507 111 Dim stDocName As String
1508 112 Dim stLinkCriteria As String
1509 113
1510 114 ' Close the current form
1511 115 DoCmd.Close
1512 116 ' Open specified form
1513 117 stDocName = "frmLogicalProperty"
1514 118 DoCmd.OpenForm stDocName, , , stLinkCriteria
1515 119
1516 120 Exit_Command4_Click
1517 121 Exit Sub
1518 122
1519 123 Err_Command4_Click
1520 124 MsgBox Err.Description
1521 125 Resume Exit_Command4_Click
1522 126
1523 127 End Sub
1524 128
1525 129 Private Sub Command6_Click()
1526 130 On Error GoTo Err_Command6_Click
1527 131
1528 132 Dim stDocName As String
1529 133 Dim stLinkCriteria As String
1530 134
1531 135 ' Close the current form
1532 136 DoCmd.Close
1533 137 ' Open specified form
1534 138 stDocName = "frmSchoolToPropertyMaint"
1535 139 DoCmd.OpenForm stDocName, , , stLinkCriteria
1536 140
1537 141 Exit_Command6_Click
1538 142 Exit Sub
1539 143
1540 144 Err_Command6_Click
1541 145 MsgBox Err.Description
1542 146 Resume Exit_Command6_Click
1543 147
1544 148 End Sub
1545 149
1546 150 Private Sub Command8_Click()
1547 151 On Error GoTo Err_Command8_Click
1548 152
1549 153 Dim stDocName As String
1550 154 Dim stLinkCriteria As String
1551 155
1552 156 ' Close the current form
1553 157 DoCmd.Close
1554 158 ' Open specified form
1555 159 stDocName = "frmRDAL_Main_Switchboard"
1556 160 DoCmd.OpenForm stDocName, , , stLinkCriteria
1557 161
1558 162 Exit_Command8_Click
1559 163 Exit Sub
1560 164

```

```

1561 165 Err_Command8_Click
1562 166 MsgBox Err.Description
1563 167 Resume Exit_Command8_Click
1564 168
1565 169 End Sub
1566
1567 Form: frmPEF
1568 Code
1569 1 Attribute VB_Name = "Form_frmPEF"
1570 2 Attribute VB_Creatable = True
1571 3 Attribute VB_PredeclaredId = True
1572 4 Attribute VB_Exposed = False
1573 5 Option Compare Database
1574 6 Option Explicit
1575 7
1576 8 Private Sub cmbPEFind_AfterUpdate()
1577 9
1578 10 Dim R As Recordset
1579 11 Set R = Me.RecordsetClone
1580 12 R.FindFirst "[PEF_PK] = " & Chr(34) & Me(cmbPEFind) & Chr(34)
1581 13 Me.Bookmark = R.Bookmark
1582 14 Me(cmbPEFind) = Null
1583 15 Me.PEF_PK.SetFocus
1584 16
1585 17 End Sub
1586 18
1587 19 Private Sub Form_BeforeUpdate(Cancel As Integer)
1588 20
1589 21 [PEF_TimeStamp] = Now
1590 22
1591 23 End Sub
1592 24
1593 25 Private Sub Form_Current()
1594 26
1595 27 Dim strSQL As String
1596 28
1597 29 ' Update the PEF List
1598 30 strSQL = "SELECT DISTINCTROW [MARINE] ([PEF]FROM MARINE LEFT JOIN PEF
1599 31 ON [MARINE] ([PEF] = [PEF].[PEF_PK]WHERE ([PEF].[PEF_PK] IS NULLORDER BY
1600 32 Me.PEF_PK.RowSource = strSQL
1601 33
1602 33 End Sub
1603 34
1604 35 Private Sub PEF_PK_AfterUpdate()
1605 36
1606 37 Dim strSQL As String
1607 38
1608 39 ' Update the PEF List
1609 40 strSQL = "SELECT DISTINCTROW [MARINE] ([PEF]FROM MARINE LEFT JOIN PEF
1610 41 ON [MARINE] ([PEF] = [PEF].[PEF_PK]WHERE ([PEF].[PEF_PK] IS NULLORDER BY
1611 42 Me.PEF_PK.RowSource = strSQL
1612 43
1613 43 End Sub
1614 44 Private Sub btnDelete_Click()
1615 45 On Error GoTo Err_btnDelete_Click
1616 46
1617 47 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 8, acMenuVer70
1618 48 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 6, acMenuVer70
1619 49
1620 50
1621 51 Exit_btnDelete_Click
1622 52 Exit Sub
1623 53
1624 54 Err_btnDelete_Click
1625 55 MsgBox Err.Description
1626 56 Resume Exit_btnDelete_Click
1627 57
1628 58 End Sub
1629 59 Private Sub btnClose_Click()
1630 60 On Error GoTo Err_btnClose_Click
1631 61
1632 62 DoCmd.Close
1633 63 DoCmd.OpenForm "frmMaintenanceSwitchboard"
1634 64
1635 65 Exit_btnClose_Click
1636 66 Exit Sub
1637 67
1638 68 Err_btnClose_Click
1639 69 MsgBox Err.Description
1640 70 Resume Exit_btnClose_Click
1641 71
1642 72 End Sub
1643 73
1644 73 End Sub
1645
1646 Form: frmPrepareAndExecuteSoler
1647 Code
1648 1 Attribute VB_Name = "Form_frmPrepareAndExecuteSoler"
1649 2 Attribute VB_Creatable = True
1650 3 Attribute VB_PredeclaredId = True
1651 4 Attribute VB_Exposed = False
1652 5 Option Compare Database
1653 6 Option Explicit
1654 7
1655 8 Private Sub btnAMPL_Click()
1656 9 On Error GoTo Err_btnAMPL_Click
1657 10
1658 11 AmpData
1659 12 Dim stAppName As String
1660 13
1661 14 stAppName = "C:\ToBeRdmAmpl\winampwin.exe"
1662 15 stAppName = "C:\ToBeRdmAmpl\winamp.bat"
1663 16 Call Shell(stAppName, 0)
1664 17
1665 18 MsgBox "Enter into the command line: 'include rdm.run'"
1666 19
1667 20 Exit_btnAMPL_Click
1668 21 Exit Sub
1669 22
1670 23 Err_btnAMPL_Click
1671 24 MsgBox Err.Description
1672 25 Resume Exit_btnAMPL_Click
1673 26
1674 26 End Sub
1675 27
1676 28 Private Sub btnAmplPlus_Click()
1677 29 On Error GoTo Err_btnAmplPlus_Click
1678 30
1679 31 AmpData
1680 32 Dim stAppName As String
1681 33
1682 34 stAppName = "C:\AMPLPLUS\AMPLPLUS.EXE rdm.ampl"
1683 35 Call Shell(stAppName, 1)
1684 36
1685 37 Exit_btnAmplPlus_Click
1686 38 Exit Sub
1687 39
1688 40 Err_btnAmplPlus_Click
1689 41 MsgBox Err.Description
1690 42

```

```

1691 45 Resume Exit_btnAmplPlus_Click
1692 46
1693 47 End Sub
1694 48
1695 49 Private Sub btnReturnPrevious_Click()
1696 50 On Error GoTo Err_btnReturnPrevious_Click
1697 51
1698 52 Dim stDocName As String
1699 53 Dim stLinkCriteria As String
1700 54
1701 55 ' Close the current form
1702 56 DoCmd.Close
1703 57
1704 58 ' Open specified form
1705 59 stDocName = "frmPreprocessing&ExecutionSwitchboard"
1706 60 DoCmd.OpenForm stDocName, , stLinkCriteria
1707 61
1708 62 Exit_btnReturnPrevious_Click
1709 63 Exit Sub
1710 64
1711 65 Err_btnReturnPrevious_Click
1712 66 MsgBox Err.Description
1713 67 Resume Exit_btnReturnPrevious_Click
1714 68
1715 69 End Sub
1716 70
1717 71 End Sub
1718 72
1719 73 Private Sub Command4_Click()
1720 74 Application.SetOption "Confirm Action Queries", False
1721 75 DoCmd.SetWarnings (False)
1722 76 DoCmd.RunSQL "DELETE FROM AMPL_RESULT;"
1723 77 Amp_Result
1724 78 Application.SetOption "Confirm Action Queries", True
1725 79 DoCmd.SetWarnings (True)
1726 80
1727 81 End Sub
1728 82
1729 83 Private Sub Command7_Click()
1730 84 On Error GoTo Err_Command7_Click
1731 85
1732 86 Dim stDocName As String
1733 87 Dim stLinkCriteria As String
1734 88
1735 89 ' Close the current form
1736 90 DoCmd.Close
1737 91 ' Open specified form
1738 92 stDocName = "frmAnalyzeResult"
1739 93 DoCmd.OpenForm stDocName, , stLinkCriteria
1740 94
1741 95 Exit_Command7_Click
1742 96 Exit Sub
1743 97
1744 98 Err_Command7_Click
1745 99 MsgBox Err.Description
1746 100 Resume Exit_Command7_Click
1747 101
1748 102 End Sub
1749
1750 Form: frmPreprocessing&ExecutionSwitchboard
1751 Code
1752 1 Attribute VB_Name = "Form_frmPreprocessing&ExecutionSwitchboard"
1753 2 Attribute VB_Creatable = True
1754 3 Attribute VB_PredeclaredId = True
1755 4 Attribute VB_Exposed = False
1756 5 Option Compare Database
1757 6 Option Explicit
1758 7
1759 8 Private Sub btnExit_Click()
1760 9 On Error GoTo Err_btnExit_Click
1761 10
1762 11 DoCmd.Close
1763 12
1764 13 Exit_btnExit_Click
1765 14 Exit Sub
1766 15
1767 16 Err_btnExit_Click
1768 17 MsgBox Err.Description
1769 18 Resume Exit_btnExit_Click
1770 19
1771 20 End Sub
1772 21
1773 22 End Sub
1774 23
1775 24 Private Sub btnReclassification_Click()
1776 25 On Error GoTo Err_btnReclassification_Click
1777 26
1778 27 Dim stDocName As String
1779 28 Dim stLinkCriteria As String
1780 29
1781 30 ' Close the current form
1782 31 DoCmd.Close
1783 32 ' Open specified form
1784 33 stDocName = "frmReclassification"
1785 34 DoCmd.OpenForm stDocName, , stLinkCriteria
1786 35
1787 36 Exit_btnReclassification_Click
1788 37 Exit Sub
1789 38
1790 39 Err_btnReclassification_Click
1791 40 MsgBox Err.Description
1792 41 Resume Exit_btnReclassification_Click
1793 42
1794 43 End Sub
1795 44
1796 45 Private Sub btnScrubMarineData_Click()
1797 46 On Error GoTo Err_btnScrubMarineData_Click
1798 47
1799 48 Dim stDocName As String
1800 49 Dim stLinkCriteria As String
1801 50
1802 51 ' Close the current form
1803 52 DoCmd.Close
1804 53 ' Open specified form
1805 54 stDocName = "frmScrubMarine"
1806 55 DoCmd.OpenForm stDocName, , stLinkCriteria
1807 56
1808 57 Exit_btnScrubMarineData_Click
1809 58 Exit Sub
1810 59
1811 60 Err_btnScrubMarineData_Click
1812 61 MsgBox Err.Description
1813 62 Resume Exit_btnScrubMarineData_Click
1814 63
1815 64 End Sub
1816 65
1817 66 Private Sub btnSpecialAssignment_Click()
1818 67 On Error GoTo Err_btnSpecialAssignment_Click
1819 68
1820 69

```



```

1821 70 Dim stDocName As String
1822 71 Dim stLinkCriteria As String
1823 72
1824 73 ' Close the current form
1825 74 DoCmd.Close
1826 75 ' Open specified form
1827 76 stDocName = "frmSpecialAssignment"
1828 77 DoCmd.OpenForm stDocName, , stLinkCriteria
1829 78
1830 79 Exit_LnkSpecialAssignment_Click
1831 80 Exit Sub
1832 81
1833 82 Err_LnkSpecialAssignment_Click
1834 83 MsgBox Err.Description
1835 84 Resume Exit_LnkSpecialAssignment_Click
1836 85
1837 86
1838 87
1839 88 End Sub
1840 89
1841 90 Private Sub Command2_Click()
1842 91 On Error GoTo Err_Command2_Click
1843 92
1844 93 Dim stDocName As String
1845 94 Dim stLinkCriteria As String
1846 95
1847 96 ' Close the current form
1848 97 DoCmd.Close
1849 98 ' Open specified form
1850 99 stDocName = "frmClassroomPenaltyAndFtr"
1851 100 DoCmd.OpenForm stDocName, , stLinkCriteria
1852 101
1853 102 Exit_Command2_Click
1854 103 Exit Sub
1855 104
1856 105 Err_Command2_Click
1857 106 MsgBox Err.Description
1858 107 Resume Exit_Command2_Click
1859 108
1860 109 End Sub
1861 110 Private Sub Command4_Click()
1862 111
1863 112 Application.SetOption "Confirm Action Queries", False
1864 113 DoCmd.SetWarnings (False)
1865 114 DoCmd.RunSQL "DELETE * FROM AMPL_RESULT;"
1866 115 Ampl_Result
1867 116 Application.SetOption "Confirm Action Queries", True
1868 117 DoCmd.SetWarnings (True)
1869 118
1870 119
1871 120 End Sub
1872 121
1873 122 Private Sub Command5_Click()
1874 123 On Error GoTo Err_Command5_Click
1875 124
1876 125 Dim stDocName As String
1877 126 Dim stLinkCriteria As String
1878 127
1879 128 ' Close the current form
1880 129 DoCmd.Close
1881 130 ' Open specified form
1882 131 stDocName = "frmPrepareAndExecuteSolver"
1883 132 DoCmd.OpenForm stDocName, , stLinkCriteria
1884 133
1885 134 Exit_Command5_Click
1886 135 Exit Sub
1887 136
1888 137 Err_Command5_Click
1889 138 MsgBox Err.Description
1890 139 Resume Exit_Command5_Click
1891 140
1892 141 End Sub
1893 142
1894 143 Private Sub Command6_Click()
1895 144 On Error GoTo Err_Command6_Click
1896 145
1897 146 Dim stDocName As String
1898 147 Dim stLinkCriteria As String
1899 148
1900 149 ' Close the current form
1901 150 DoCmd.Close
1902 151 ' Open specified form
1903 152 stDocName = "frmRDM_Main_Switchboard"
1904 153 DoCmd.OpenForm stDocName, , stLinkCriteria
1905 154
1906 155 Exit_Command6_Click
1907 156 Exit Sub
1908 157
1909 158 Err_Command6_Click
1910 159 MsgBox Err.Description
1911 160 Resume Exit_Command6_Click
1912 161
1913 162 End Sub
1914
1915 Form: frmRDM_Main_Switchboard
1916 Code
1917 1 Attribute VB_Name = "Form_frmRDM_Main_Switchboard"
1918 2 Attribute VB_Creatable = True
1919 3 Attribute VB_PredeclaredId = True
1920 4 Attribute VB_Exposed = False
1921 5 Option Compare Database
1922 6 Option Explicit
1923 7 Private Sub Command0_Click()
1924 8
1925 9 On Error GoTo Err_Command0_Click
1926 10
1927 11 Dim stDocName As String
1928 12 Dim stLinkCriteria As String
1929 13
1930 14 ' Close the current form
1931 15 DoCmd.Close
1932 16 ' Open specified form
1933 17 stDocName = "frmImportExportSwitchboard"
1934 18 DoCmd.OpenForm stDocName, , stLinkCriteria
1935 19
1936 20 Exit_Command0_Click
1937 21 Exit Sub
1938 22
1939 23 Err_Command0_Click
1940 24 MsgBox Err.Description
1941 25 Resume Exit_Command0_Click
1942 26
1943 27 End Sub
1944 28 Private Sub Command2_Click()
1945 29 On Error GoTo Err_Command2_Click
1946 30
1947 31 Dim stDocName As String
1948 32 Dim stLinkCriteria As String
1949 33
1950 34 ' Close the current form

```

```

1951 35 DoCmd.Close
1952 36 ' Open specified form
1953 37 stDocName = "frmMaintenanceSwitchboard"
1954 38 DoCmd.OpenForm stDocName, , stLinkCriteria
1955 39
1956 40 Exit_Command2_Click
1957 41 Exit Sub
1958 42
1959 43 Err_Command2_Click
1960 44 MsgBox Err.Description
1961 45 Resume Exit_Command2_Click
1962 46
1963 47 End Sub
1964 48 Private Sub Command4_Click()
1965 49 On Error GoTo Err_Command4_Click
1966 50
1967 51 Dim stDocName As String
1968 52 Dim stLinkCriteria As String
1969 53
1970 54 ' Close the current form
1971 55 DoCmd.Close
1972 56 ' Open specified form
1973 57 stDocName = "frmPreprocessingExecutionSwitchboard"
1974 58 DoCmd.OpenForm stDocName, , stLinkCriteria
1975 59
1976 60 Exit_Command4_Click
1977 61 Exit Sub
1978 62
1979 63 Err_Command4_Click
1980 64 MsgBox Err.Description
1981 65 Resume Exit_Command4_Click
1982 66
1983 67 End Sub
1984 68 Private Sub Command6_Click()
1985 69 On Error GoTo Err_Command6_Click
1986 70
1987 71 Dim stDocName As String
1988 72 Dim stLinkCriteria As String
1989 73
1990 74 ' Close the current form
1991 75 DoCmd.Close
1992 76 ' Open specified form
1993 77 stDocName = "frmAnalyzeResult"
1994 78 DoCmd.OpenForm stDocName, , stLinkCriteria
1995 79
1996 80 Exit_Command6_Click
1997 81 Exit Sub
1998 82
1999 83 Err_Command6_Click
2000 84 MsgBox Err.Description
2001 85 Resume Exit_Command6_Click
2002 86
2003 87 End Sub
2004 88 Private Sub Command8_Click()
2005 89 On Error GoTo Err_Command8_Click
2006 90
2007 91 Dim stDocName As String
2008 92 Dim stLinkCriteria As String
2009 93
2010 94 ' Close the current form
2011 95 DoCmd.Close
2012 96 ' Open specified form
2013 97 stDocName = "frmGenerateReport"
2014 98 DoCmd.OpenForm stDocName, , stLinkCriteria
2015 99
2016 100 Exit_Command8_Click
2017 101 Exit Sub
2018 102
2019 103 Err_Command8_Click
2020 104 MsgBox Err.Description
2021 105 Resume Exit_Command8_Click
2022 106
2023 107 End Sub
2024 108 Private Sub btnExit_Click()
2025 109 On Error GoTo Err_btnExit_Click
2026 110
2027 111
2028 112 DoCmd.Close
2029 113
2030 114 Exit_UnlnExit_Click
2031 115 Exit Sub
2032 116
2033 117 Err_btnExit_Click
2034 118 MsgBox Err.Description
2035 119 Resume Exit_UnlnExit_Click
2036 120
2037 121 End Sub
2038
2039 Form: frmRecClassification
2040 Code
2041 1 Attribute VB_Name = "Form_frmRecClassification"
2042 2 Attribute VB_Creatable = True
2043 3 Attribute VB_PredeclaredId = True
2044 4 Attribute VB_Exposed = False
2045 5 Option Compare Database
2046 6 Option Explicit
2047 7
2048 8 Private Sub cmbClassNumber_AfterUpdate()
2049 9
2050 10 Dim strConvert As String
2051 11
2052 12 Me.txtReportDate = Me.cmbClassNumber.Column(2)
2053 13 Me.txtMCC = Me.cmbClassNumber.Column(4)
2054 14
2055 15 ' Convert the fiscal year to a two digit number for RDS file
2056 16 strConvert = CStr(Me.cmbClassNumber.Column(3))
2057 17 strConvert = Right(strConvert, 2)
2058 18 Me.txtFY = CInt(strConvert)
2059 19
2060 20 Me.txtAssignmentType = "R"
2061 21
2062 22 End Sub
2063 23
2064 24 Private Sub cmbClassNumber_Enter()
2065 25
2066 26 Dim strSQL As String
2067 27
2068 28 ' This query finds the class numbers, report dates and class
2069 29 covering dates associated with the chosen school
2070 30 strSQL = "SELECT BNA_EXTRACT.ClassNumber_PK,
2071 BNA_EXTRACT.ClassConvDate, BNA_EXTRACT.ReportDate,
2072 BNA_EXTRACT.FiscalYear_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT INNER JOIN
2073 SCH_TGT_MOS ON (BNA_EXTRACT.CourseNumber_PK =
2074 SCH_TGT_MOS.CourseNumber_PK) AND (BNA_EXTRACT.TargetMOS_PK =
2075 SCH_TGT_MOS.TargetMOS_PK) WHERE BNA_EXTRACT.CourseNumber_PK =
2076 Form!frmRecClassification!cmbCourseNumber AND SCH_TGT_MOS.AMOS_FK =
2077 Form!frmRecClassification!txtAMOS;"
2078 30 Me.cmbClassNumber.RowSource = strSQL
2079 31
2080 32 End Sub

```

```

2081 33 Private Sub cmbCourseNumber_AfterUpdate()
2082 34 Private Sub cmbCourseNumber_AfterUpdate()
2083 35
2084 36 Me.bfAMOS = Me.cmbCourseNumber.Column(1)
2085 37
2086 38 End Sub
2087 39
2088 40 Private Sub cmbSSN_AfterUpdate()
2089 41
2090 42 Me.bfSCI = Me.cmbSSN.Column(1)
2091 43 Me.bfGradDate = Me.cmbSSN.Column(2)
2092 44
2093 45 End Sub
2094 46
2095 47 Private Sub cmbSSN_Enter()
2096 48
2097 49 Dim strSQL As String
2098 50
2099 51 strSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SCI,
2100 MARINE.GradDate FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
2101 ASSIGNMENT.SSN_FK WHERE (((ASSIGNMENT.SSN_FK) Is Null));"
2102 52 Me.cmbSSN.RowSource = strSQL
2103 53
2104 54 End Sub
2105 55
2106 56 Private Sub cmbSSNFind_AfterUpdate()
2107 57
2108 58 Dim R As Recordset
2109 59 Set R = Me.RecordsetClone
2110 60 R.FindFirst "[SSN_PK] = " & Chr(34) & Me.cmbSSNFind & Chr(34)
2111 61 Me.Bookmark = R.Bookmark
2112 62 Me.cmbSSNFind = Null
2113 63 Me.cmbSSN.SetFocus
2114 64
2115 65 End Sub
2116 66
2117 67 Private Sub Delete_Click()
2118 68 On Error GoTo Err_Delete_Click
2119 69
2120 70
2121 71 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2122 72 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2123 73
2124 74 Exit_Delete_Click:
2125 75 Exit Sub
2126 76
2127 77 Err_Delete_Click:
2128 78 MsgBox Err.Description
2129 79 Resume Exit_Delete_Click
2130 80
2131 81 End Sub
2132 82
2133 83 Private Sub Close_Click()
2134 84 On Error GoTo Err_Close_Click
2135 85
2136 86 Dim stDocName As String
2137 87 Dim stLinkCriteria As String
2138 88
2139 89 ' Close current form
2140 90 DoCmd.Close
2141 91
2142 92 ' Open specified form
2143 93 stDocName = "frmProcessing&ExecutionSwitchboard"
2144 94 DoCmd.OpenForm stDocName, , stLinkCriteria
2145 95
2146 96 Exit_Close_Click:
2147 97 Exit Sub
2148 98
2149 99 Err_Close_Click:
2150 100 MsgBox Err.Description
2151 101 Resume Exit_Close_Click
2152 102
2153 103 End Sub
2154 104
2155 105
2156 106 Form: frmSchoolAssignments
2157 107
2158 108 1 Attribute VB_Name = "Form:frmSchoolAssignments"
2159 109 2 Attribute VB_Creatable = True
2160 110 3 Attribute VB_PredeclaredId = True
2161 111 4 Attribute VB_Exposed = False
2162 112 5 Option Compare Database
2163 113 6 Option Explicit
2164 114 7 Private Sub Close_Click()
2165 115 8 On Error GoTo Err_Close_Click
2166 116 9
2167 117 10 Dim stDocName As String
2168 118 11 Dim stLinkCriteria As String
2169 119 12 Dim db1 As Database
2170 120 13
2171 121 14 Set db1 = CurrentDb()
2172 122 15
2173 123 16 ' Close the current form
2174 124 17 DoCmd.Close
2175 125 18 ' Open specified form
2176 126 19 stDocName = "frmAnalyzeResult"
2177 127 20 DoCmd.OpenForm stDocName, , stLinkCriteria
2178 128 21 Form!frmAnalyzeResult.SetFocus
2179 129 22
2180 130 23
2181 131 24 Exit_Close_Click:
2182 132 25 Exit Sub
2183 133 26
2184 134 27 Err_Close_Click:
2185 135 28 MsgBox Err.Description
2186 136 29 Resume Exit_Close_Click
2187 137 30
2188 138 31 End Sub
2189 139 32
2190 140 33 Private Sub Form_Current()
2191 141 34
2192 142 35 Dim strSQL As String
2193 143 36 Dim rec As Recordset
2194 144 37 Dim db1 As Database
2195 145 38
2196 146 39 Set db1 = CurrentDb()
2197 147 40
2198 148 41 ' Calculates the total quota, and number of schools for the run.
2199 149 42 strSQL = "SELECT Sum(SumOfQuota) AS TotalQuota, Count(SumOfQuota) AS
2200 NumberOfSchools FROM qryTotalQuotaForRun;"
2201 150 43 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2202 151 44 Me.bfTotalQuota = rec.TotalQuota
2203 152 45 Me.bfNumberOfSchools = rec.NumberOfSchools
2204 153 46
2205 154 47 ' Calculates the total number of assignments for the run.
2206 155 48 strSQL = "SELECT Sum(CountOfFiscalYear_PK) AS TotalAssigned FROM
2207 qryTotalQuotaFiledForRun;"
2208 156 49 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2209 157 50 Me.bfTotalAssigned = rec.TotalAssigned
2210 158 51
2211 52 Me.bfTotalPercentFill = Int((Me.bfTotalAssigned / Me.bfTotalQuota)
2212 53
2213 54
2214 55 End Sub
2215 56
2216 57 Form: frmSchools
2217 58
2218 59 1 Attribute VB_Name = "Form:frmSchools"
2219 60 2 Attribute VB_Creatable = True
2220 61 3 Attribute VB_PredeclaredId = True
2221 62 4 Attribute VB_Exposed = False
2222 63 5 Option Compare Database
2223 64 6 Option Explicit
2224 65 7
2225 66 8 Private Sub AMOS_PK_AfterUpdate()
2226 67
2227 68 Dim rec As Recordset
2228 69 Dim db1 As Database
2229 70 Dim strSQL As String
2230 71
2231 72 ' Update the drop down list for AMOS
2232 73 strSQL = "SELECT TargetMOS_PK FROM
2233 qryListUnusedTargetMOSFromTARGET_MOS WHERE TCourseNumber_PK = " &
2234 Me.AMOS_PK.RowSource = strSQL
2235 74 Me.lstTargetMOS_PK.RowSource = strSQL
2236 75
2237 76 ' Update the AMOS_FK in the child relation SCH_TGT_MOS
2238 77 ' Form!frmSchools.subfrmSchools.AMOS_FK.Value = Me.AMOS_PK.Value
2239 78
2240 79 ' Update the TargetMOS_FK in the child relation SCH_TGT_MOS
2241 80 ' Set db1 = CurrentDb()
2242 81 ' Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2243 82 ' Form!frmSchools.subfrmSchools.TargetMOS_FK.Value = rec.TargetMOS_PK
2244 83
2245 84 ' Input the correct MCC value for the selected course
2246 85 strSQL = "SELECT MCC FROM TARGET_MOS WHERE TCourseNumber_PK = " &
2247 Me.SCourseNumber_PK.Value & ""
2248 86 Set rec = CurrentDb()
2249 87 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2250 88 Me.MCC.Value = rec.MCC
2251 89
2252 90 ' Update the TCourseNumber_FK in the child relation SCH_TGT_MOS
2253 91 ' Form!frmSchools.subfrmSchools.TCourseNumber_FK.Value =
2254 92 Me.SCourseNumber_PK.Value
2255 93
2256 94 End Sub
2257 95
2258 96 Private Sub btnClear_Click()
2259 97
2260 98 Dim db1 As Database
2261 99 Dim strSQL As String
2262 100 Dim i As Integer
2263 101 Dim rec As Recordset
2264 102
2265 103 Set db1 = CurrentDb()
2266 104
2267 105 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2268 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2269 SCH_TGT_MOS.TargetMOS_FK AND TARGET_MOS.TCourseNumber_PK =
2270 SCH_TGT_MOS.TCourseNumber_FK WHERE (((SCH_TGT_MOS.TargetMOS_FK) Is
2271 Null) AND ((SCH_TGT_MOS.TCourseNumber_FK) Is Null)) ORDER BY
2272 'TCourseNumber_PK'"
2273 106 Set rec = db1.OpenRecordset(strSQL, dbOpenDynaset)
2274 107
2275 108 If rec.EOF = False Then
2276 109 rec.MoveNext
2277 110
2278 111 End If
2279 112
2280 113 ' Clear unused course numbers from the target_mos table
2281 114
2282 115 If rec.RecordCount > 0 Then
2283 116 For i = 1 To rec.RecordCount
2284 117 ' Disables the action query confirmation message
2285 118 Application.SetOption "Confirm Action Queries", False
2286 119 DoCmd.SetWarnings (False)
2287 120
2288 121 strSQL = "DELETE * FROM TARGET_MOS WHERE
2289 TARGET_MOS.TCourseNumber_PK = " & rec.TCourseNumber_PK & ""
2290 122 db1.Execute strSQL
2291 123
2292 124 ' Enables the action query confirmation message
2293 125 Application.SetOption "Confirm Action Queries", True
2294 126 DoCmd.SetWarnings (True)
2295 127 rec.MoveNext
2296 128 Next i
2297 129 End If
2298 130
2299 131 ' Update the Course Number List
2300 132 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2301 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2302 SCH_TGT_MOS.TargetMOS_FK AND TARGET_MOS.TCourseNumber_PK =
2303 SCH_TGT_MOS.TCourseNumber_FK WHERE (((SCH_TGT_MOS.TargetMOS_FK) Is
2304 Null) AND ((SCH_TGT_MOS.TCourseNumber_FK) Is Null)) ORDER BY
2305 'TCourseNumber_PK'"
2306 133 Me.SCourseNumber_PK.RowSource = strSQL
2307 134
2308 135 End Sub
2309 136
2310 137 Private Sub btnResetAllPenalties_Click()
2311 138
2312 139 ' Disables the action query confirmation message
2313 140 Application.SetOption "Confirm Action Queries", False
2314 141 DoCmd.SetWarnings (False)
2315 142
2316 143 DoCmd.RunSQL "UPDATE SCHOOL SET PenaltyFactor = 24;"
2317 144
2318 145 ' Enables the action query confirmation message
2319 146 Application.SetOption "Confirm Action Queries", True
2320 147 DoCmd.SetWarnings (True)
2321 148
2322 149 cmbPenaltyView = "1x"
2323 150
2324 151 End Sub
2325 152
2326 153 Private Sub cmbCourseNumberFind_AfterUpdate()
2327 154
2328 155 Dim R As Recordset
2329 156 Set R = Me.RecordsetClone
2330 157 R.FindFirst "[SCourseNumber_PK] = " & Chr(34) &
2331 158 Me.cmbCourseNumberFind & Chr(34)
2332 159 Me.Bookmark = R.Bookmark
2333 160 Me.cmbCourseNumberFind = Null
2334 161 Me.SCourseNumber_PK.SetFocus
2335 162
2336 163 End Sub
2337 164
2338 165 Private Sub cmbPenaltyView_AfterUpdate()
2339 166
2340 167

```

```

2340 108 Select Case cmbPenaltyView
2341 109 Case "4x"
2342 110 bPenaltyFactor = 4 * 24
2343 111 Case "3x"
2344 112 bPenaltyFactor = 3 * 24
2345 113 Case "2x"
2346 114 bPenaltyFactor = 2 * 24
2347 115 Case "1x"
2348 116 bPenaltyFactor = 1 * 24
2349 117 Case "1/2x"
2350 118 bPenaltyFactor = 24 / 2
2351 119 Case "1/3x"
2352 120 bPenaltyFactor = 24 / 3
2353 121 Case "1/4x"
2354 122 bPenaltyFactor = 24 / 4
2355 123 End Select
2356 124
2357 125 ' Save the update
2358 126 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord,
2359 127
2360 128 End Sub
2361 129
2362 130 Private Sub Command14_Click()
2363 131
2364 132 Dim strSQL As String
2365 133
2366 134 ' Disables the action query confirmation message
2367 135 Application.SetOption "Confirm Action Queries", False
2368 136 DoCmd.SetWarnings (False)
2369 137
2370 138 DoCmd.RunSQL "INSERT INTO TARGET_MOS ( TCourseNumber_PK,
2371 TargetMOS_PK, MCC ) SELECT DISTINCT BNA_EXTRACT.CourseNumber_PK,
2372 BNA_EXTRACT.TargetMOS_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT;"
2373 139
2374 140 ' Enables the action query confirmation message
2375 141 Application.SetOption "Confirm Action Queries", True
2376 142 DoCmd.SetWarnings (True)
2377 143
2378 144 ' Update the Course Number List
2379 145 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2380 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2381 SCH_TGT_MOS.TargetMOS_PK AND TARGET_MOS.TCourseNumber_PK =
2382 SCH_TGT_MOS.TCourseNumber_PK WHERE ((SCH_TGT_MOS.TargetMOS_PK Is
2383 Null)AND(SCH_TGT_MOS.TCourseNumber_PK Is Null)ORDER BY
2384 TCourseNumber_PK;"
2385 146 Me.SCourseNumber_PK.RowSource = strSQL
2386 147
2387 148 End Sub
2388 149
2389 150
2390 151 Private Sub Command30_Click()
2391 152 Dim frm As Form, c1 As Control
2392 153 Dim varItem As Variant, int1 As Integer
2393 154 Dim strSQL As String
2394 155
2395 156 ' Ensures the current record is save to the SCHOOL table
2396 157 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord,
2397 158
2398 159 ' This code enters the selected Target MOS's into the SCH_TGT_MOS
2399 160 Set frm = Forms!frmSchoole
2400 161 Set c1 = frm!tblTargetMOS_FK
2401 162 For Each varItem In c1.ItemsSelected
2402 163 For int1 = 0 To c1.ColumnCount - 1
2403 164
2404 165 ' This puts the value found in the list box into a text box,
2405 166 making it readable by the query
2406 167 Forms!frmSchoole.TargetValue.Value = c1.Column(int1, varItem)
2407 168
2408 169 ' Disables the action query confirmation message
2409 170 Application.SetOption "Confirm Action Queries", False
2410 171
2411 172 ' Enters the value into the SCH_TGT_MOS table
2412 173 DoCmd.OpenQuery "qryUpdateSCH_TGT_MOS"
2413 174
2414 175 ' Enables the action query confirmation message
2415 176 Application.SetOption "Confirm Action Queries", True
2416 177
2417 178 Next int1
2418 179 Next varItem
2419 180
2420 181 ' Update the Target MOS list
2421 182 strSQL = "SELECT (TargetMOS_FK) FROM (SCH_TGT_MOS) WHERE
2422 [SCourseNumber_PK] = " & Forms!frmSchoole!SCourseNumber_PK.Value & " " &
2423 " AND AMOS_PK = " & Forms!frmSchoole!AMOS_PK.Value & " " &
2424 " Form!frmSchoole!tblTargetMOS_FK_History.RowSource = strSQL
2425 183
2426 184 ' Update the Potential Target MOS list
2427 185 strSQL = "SELECT TargetMOS_PK FROM
2428 qryListUnusedTargetMOSFromTARGET_MOS WHERE TCourseNumber_PK = " &
2429 Me.LstTargetMOS_FK.RowSource = strSQL
2430 186
2431 187 ' Update the Course Number List
2432 188 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2433 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2434 SCH_TGT_MOS.TargetMOS_PK WHERE ((SCH_TGT_MOS.TargetMOS_PK Is
2435 Null)AND(SCH_TGT_MOS.TCourseNumber_PK Is Null)ORDER BY
2436 TCourseNumber_PK;"
2437 189
2438 190 ' Update time stamp
2439 191 [Sch_TimeStamp] = Now
2440 192
2441 193 End Sub
2442 194
2443 195 Private Sub Command40_Click()
2444 196 Dim frm As Form, c1 As Control
2445 200 Dim varItem As Variant, int1 As Integer
2446 201 Dim strSQL As String
2447 202
2448 203 ' This code Deletes the selected Target MOS's and its associated
2449 204 entries from the SCH_TGT_MOS table
2450 205 Set frm = Forms!frmSchoole
2451 206 Set c1 = frm!tblTargetMOS_FK_History
2452 207 For Each varItem In c1.ItemsSelected
2453 208 For int1 = 0 To c1.ColumnCount - 1
2454 209
2455 210 ' This puts the value found in the list box into a text box,
2456 211 making it readable by the query
2457 212 Forms!frmSchoole.TargetValue.Value = c1.Column(int1, varItem)
2458 213
2459 214 ' Disables the action query confirmation message
2460 215 Application.SetOption "Confirm Action Queries", False
2461 216
2462 217 ' Deletes the record associated with the specified value from
2463 218 the SCH_TGT_MOS table
2464 219 DoCmd.OpenQuery "qryDeleteSCH_TGT_MOS"
2465 220
2466 221 ' Enables the action query confirmation message
2467 222 Application.SetOption "Confirm Action Queries", True
2468 223
2469 224 Next int1
2470 225
2471 226 Next varItem
2472 227
2473 228 ' Update the Target MOS list
2474 229 strSQL = "SELECT (TargetMOS_FK) FROM (SCH_TGT_MOS) WHERE
2475 230 [SCourseNumber_PK] = " & Forms!frmSchoole!SCourseNumber_PK.Value & " " &
2476 231 " AND AMOS_PK = " & Forms!frmSchoole!AMOS_PK.Value & " " &
2477 232 " Form!frmSchoole!tblTargetMOS_FK_History.RowSource = strSQL
2478 233
2479 234 ' Update the Potential Target MOS list
2480 235 strSQL = "SELECT TargetMOS_PK FROM
2481 236 qryListUnusedTargetMOSFromTARGET_MOS WHERE TCourseNumber_PK = " &
2482 237 Me.LstTargetMOS_FK.RowSource = strSQL
2483 238
2484 239 ' Update the Course Number List
2485 240 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2486 241 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2487 242 SCH_TGT_MOS.TargetMOS_PK AND TARGET_MOS.TCourseNumber_PK =
2488 243 SCH_TGT_MOS.TCourseNumber_PK WHERE ((SCH_TGT_MOS.TargetMOS_PK Is
2489 244 Null)AND(SCH_TGT_MOS.TCourseNumber_PK Is Null)ORDER BY
2490 245 TCourseNumber_PK;"
2491 246
2492 247 Me.SCourseNumber_PK.RowSource = strSQL
2493 248
2494 249 ' Forces user to enter date for course number, before the Assigned
2495 250 If IsNull(Me.SCourseNumber_PK.Value) Then
2496 251 Me!AMOS_PK.Enabled = False
2497 252
2498 253 ' Allows the user to enter an assigned MOS
2499 254 Else
2500 255 Me!AMOS_PK.Enabled = True
2501 256 End If
2502 257
2503 258 ' Prevents the user from accidentally changing a course number
2504 259 If IsNull(Me.SCourseNumber_PK) Then
2505 260 Me!SCourseNumber_PK.Locked = False
2506 261
2507 262 Else
2508 263 Me!SCourseNumber_PK.Locked = True
2509 264 End If
2510 265
2511 266 ' Prevents the user from accidentally changing an AMOS
2512 267 If IsNull(AMOS_PK) Then
2513 268 Me!AMOS_PK.Locked = False
2514 269
2515 270 Else
2516 271 Me!AMOS_PK.Locked = True
2517 272 End If
2518 273
2519 274 ' Update the penalty factor view
2520 275 Select Case bPenaltyFactor
2521 276 Case 06
2522 277 cmbPenaltyView = "4x"
2523 278
2524 279 Case 72
2525 280 cmbPenaltyView = "3x"
2526 281
2527 282 Case 48
2528 283 cmbPenaltyView = "2x"
2529 284
2530 285 Case 24
2531 286 cmbPenaltyView = "1x"
2532 287
2533 288 Case 12
2534 289 cmbPenaltyView = "1/2x"
2535 290
2536 291 Case 8
2537 292 cmbPenaltyView = "1/3x"
2538 293
2539 294 Case 6
2540 295 cmbPenaltyView = "1/4x"
2541 296
2542 297 End Select
2543 298
2544 299 End Sub
2545 300
2546 301 Private Sub btnDelete_Click()
2547 302 On Error GoTo Err_btnDelete_Click
2548 303
2549 304 ' Disables the action query confirmation message
2550 305 Application.SetOption "Confirm Action Queries", False
2551 306
2552 307 ' Deletes the record associated with the specified value from the
2553 308 SCH_TGT_MOS table
2554 309 DoCmd.OpenQuery "qryDeleteSCH_TGT_MOSEntry"
2555 310
2556 311 ' Deletes the record associated with the specified value from the
2557 312 SCHOOL table
2558 313 DoCmd.OpenQuery "qryDeleteSCHOOLEntry"
2559 314
2560 315 ' Enables the action query confirmation message
2561 316
2562 317
2563 318
2564 319
2565 320
2566 321
2567 322
2568 323
2569 324
2570 325
2571 326
2572 327
2573 328
2574 329
2575 330
2576 331
2577 332
2578 333
2579 334
2580 335

```

```

2600 Application.SetOption "Confirm Action Queries", True
2601
2602 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2603 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2604
2605 341 Exit btnDelete_Click
2606
2607 343
2608 344 Err_blnDelete_Click
2609
2610 345 MsgBox Err.Description
2611 Resume Exit_blnDelete_Click
2612
2613 346 End Sub
2614
2615 350 On Error GoTo Err_blnClose_Click
2616
2617 351
2618 352
2619 353 DoCmd.Close
2620 354 DoCmd.OpenForm "frmMaintenanceSwitchboard"
2621
2622 355 Exit_blnClose_Click
2623 Exit Sub
2624
2625 359 Err_blnClose_Click
2626
2627 360 MsgBox Err.Description
2628 Resume Exit_blnClose_Click
2629
2630 361 Resume Exit_blnClose_Click
2631
2632 362 End Sub
2633
2634 Form: frmSchoolToPEF_Maint
2635 Code
2636 1 Attribute VB_Name = "Form_frmSchoolToPEF_Maint"
2637 2 Attribute VB_Creatable = True
2638 3 Attribute VB_PredeclaredId = True
2639 4 Attribute VB_Exposed = False
2640 5 Option Compare Database
2641 6 Option Explicit
2642
2643 8 Private Sub btnClose_Click()
2644 On Error GoTo Err_blnClose_Click
2645
2646 11 DoCmd.Close
2647 12 DoCmd.OpenForm "frmMaintenanceSwitchboard"
2648
2649 14 Exit_blnClose_Click
2650 Exit Sub
2651
2652 16 Err_blnClose_Click
2653
2654 17 MsgBox Err.Description
2655 Resume Exit_blnClose_Click
2656
2657 18 Err_blnClose_Click
2658
2659 19 Resume Exit_blnClose_Click
2660
2661 21
2662 22 End Sub
2663
2664 23 Private Sub btnDelete_Click()
2665 On Error GoTo Err_blnDelete_Click
2666
2667 26 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2668 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2669
2670 30 Exit_blnDelete_Click
2671 Exit Sub
2672
2673 32 Err_blnDelete_Click
2674
2675 33 MsgBox Err.Description
2676 Resume Exit_blnDelete_Click
2677
2678 35 End Sub
2679
2680 40 Private Sub cmbCourseNumberFind_AfterUpdate()
2681
2682 42 Dim R As Recordset
2683 Set R = Me.RecordsetClone
2684 R.FindFirst "[CourseNumber_PK] = " & Chr(34) &
2685 Me[cmbCourseNumberFind] & Chr(34)
2686
2687 46 Me.Bookmark = R.Bookmark
2688
2689 47 Me[cmbCourseNumberFind] = Null
2690 Me[cmbCourseNumber_PKSelfFocus]
2691
2692 50 End Sub
2693
2694 51 Private Sub Form_Current()
2695
2696 53 Dim strSQL As String
2697
2698 54 ' Update the Potential PEF list
2699 strSQL = "SELECT DISTINCTROW [PEF].[PEF_PK] FROM PEF LEFT JOIN
2700 qrySch_Pef ON ([PEF].[PEF_PK] = [qrySch_Pef].[PEF_PK] WHERE
2701 ([qrySch_Pef].[PEF_PK] Is Null) ORDER BY [PEF].[PEF_PK]"
2702 Me.lstUnselectedPEF.RowSource = strSQL
2703
2704 56 ' Update the Selected PEF List
2705 strSQL = "SELECT [PEF_PK] FROM qrySch_Pef"
2706 Me.lstSelectedPEF.RowSource = strSQL
2707
2708 62 End Sub
2709
2710 64 Private Sub btnLeft_Click()
2711 Dim frm As Form, ctl As Control
2712 Dim varIn As Variant, int As Integer
2713 Dim strSQL As String
2714
2715 ' Ensures the current record is save to the SCH_TGT table
2716 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
2717
2718 ' This code enters the selected Target MOS's into the SCH_TGT_MOS
2719 Set ctl = frm.lstUnselectedPEF
2720 For Each varIn In ctl.ItemsSelected
2721 For int = 0 To ctl.ColumnCount - 1
2722 ' This puts the value found in the list box into a text box,
2723 making it readable by the query
2724 Form!frmSchoolToPEF_Maint!PEFValue.Value = ctl.Column(int),
2725
2726 ' Disables the action query confirmation message
2727 Application.SetOption "Confirm Action Queries", False
2728
2729 ' Enters the value into the SCH_TGT_MOS table
2730 DoCmd.OpenQuery "qryUpdateSCH_PEF"
2731
2732 ' Enables the action query confirmation message
2733 Application.SetOption "Confirm Action Queries", True
2734
2735 Next int
2736 Next varIn
2737
2738 ' Update the Potential PEF list
2739 strSQL = "SELECT DISTINCTROW [PEF].[PEF_PK] FROM PEF LEFT JOIN

```

```

2730 qrySch_Pef ON ([PEF].[PEF_PK] = [qrySch_Pef].[PEF_PK] WHERE
2731 ([qrySch_Pef].[PEF_PK] Is Null) ORDER BY [PEF].[PEF_PK]"
2732 Me.lstUnselectedPEF.RowSource = strSQL
2733
2734 ' Update the Selected PEF List
2735 strSQL = "SELECT [PEF_PK] FROM qrySch_Pef"
2736 Me.lstSelectedPEF.RowSource = strSQL
2737
2738 102 End Sub
2739
2740 104 Private Sub btnRight_Click()
2741 Dim frm As Form, ctl As Control
2742 Dim varIn As Variant, int As Integer
2743 Dim strSQL As String
2744
2745 ' This code Deletes the selected Target MOS's and its associated
2746 entries from the SCH_TGT_MOS table
2747 Set frm = Form!frmSchoolToPEF_Maint
2748 Set ctl = frm.lstSelectedPEF
2749 For Each varIn In ctl.ItemsSelected
2750 For int = 0 To ctl.ColumnCount - 1
2751 ' This puts the value found in the list box into a text box,
2752 making it readable by the query
2753 Form!frmSchoolToPEF_Maint!PEFValue.Value = ctl.Column(int),
2754
2755 ' Disables the action query confirmation message
2756 Application.SetOption "Confirm Action Queries", False
2757
2758 ' Deletes the record associated with the specified value from
2759 the SCH_TGT_MOS table
2760 DoCmd.OpenQuery "qryDeleteSCH_PEF"
2761
2762 ' Enables the action query confirmation message
2763 Application.SetOption "Confirm Action Queries", True
2764
2765 Next int
2766 Next varIn
2767
2768 ' Update the Potential PEF list
2769 strSQL = "SELECT DISTINCTROW [PEF].[PEF_PK] FROM PEF LEFT JOIN
2770 qrySch_Pef ON ([PEF].[PEF_PK] = [qrySch_Pef].[PEF_PK] WHERE
2771 ([qrySch_Pef].[PEF_PK] Is Null) ORDER BY [PEF].[PEF_PK]"
2772 Me.lstUnselectedPEF.RowSource = strSQL
2773
2774 ' Update the Selected PEF List
2775 strSQL = "SELECT [PEF_PK] FROM qrySch_Pef"
2776 Me.lstSelectedPEF.RowSource = strSQL
2777
2778 137 End Sub
2779
2780 Form: frmSchoolToPropertyMaint
2781 Code
2782 1 Attribute VB_Name = "Form_frmSchoolToPropertyMaint"
2783 2 Attribute VB_Creatable = True
2784 3 Attribute VB_PredeclaredId = True
2785 4 Attribute VB_Exposed = False
2786 5 Option Compare Database
2787 6 Option Explicit
2788
2789 8 Private Sub btnClose_Click()
2790 On Error GoTo Err_blnClose_Click
2791
2792 10 DoCmd.Close
2793 11 DoCmd.OpenForm "frmMaintenanceSwitchboard"
2794
2795 13 Exit_blnClose_Click
2796 Exit Sub
2797
2798 15 Err_blnClose_Click
2799
2800 16 MsgBox Err.Description
2801 Resume Exit_blnClose_Click
2802
2803 18 End Sub
2804
2805 20 Private Sub btnDelete_Click()
2806 On Error GoTo Err_blnDelete_Click
2807
2808 22 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2809 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2810
2811 26 Exit_blnDelete_Click
2812 Exit Sub
2813
2814 28 Err_blnDelete_Click
2815
2816 29 MsgBox Err.Description
2817 Resume Exit_blnDelete_Click
2818
2819 31 End Sub
2820
2821 33 Private Sub btnLevel0_Click()
2822 On Error GoTo Err_blnLevel0_Click
2823
2824 35 Dim lngRecordNum As Long
2825 Dim strSQL As String
2826 lngRecordNum = Me.CurrentRecord
2827
2828 DoCmd.Close
2829
2830 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2831 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2832 lngRecordNum
2833 Form!frmSchoolToPropLevelMaint!ChosenLevel = 0
2834
2835 ' Update the selected property list
2836 strSQL = "SELECT [Property Name_PK] FROM FUND_SCH_PROP WHERE
2837 [Property Name_PK] = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & " And
2838 AMOS_PK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & " And
2839 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2840 UNION SELECT ([Property Name_PK] FROM LOG_SCH_PROP WHERE [Source Number_PK]
2841 = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & " And
2842 And AMOS_PK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & " And
2843 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2844 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
2845
2846 58 End Sub
2847
2848 60 Exit_blnLevel0_Click
2849 Exit Sub
2850
2851 62 Err_blnLevel0_Click
2852
2853 63 MsgBox Err.Description
2854 Resume Exit_blnLevel0_Click
2855
2856 65 End Sub
2857
2858 67 End Sub
2859

```

```

2660 69 Private Sub btnLevel1_Click()
2661 70 On Error GoTo Err_btnLevel1_Click
2662 71
2663 72 Dim lngRecordNum As Long
2664 73 Dim strSQL As String
2665 74
2666 75 lngRecordNum = Me.CurrentRecord
2667 76
2668 77 DoCmd.Close
2669 78
2670 79 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2671 80 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2672 lngRecordNum
2673 81 Form!frmSchoolToPropLevelMaint!ChosenLevel = 1
2674 82
2675 83 'Update the selected property list
2676 84 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
2677 SCourseNumber_RK = " &
2678 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & " And
2679 AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2680 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2681 UNION SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK
2682 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & "
2683 And AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2684 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2685 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
2686 85
2687 86 Exit btnLevel1_Click
2688 87 Exit Sub
2689 88
2690 89 Err_btnLevel1_Click
2691 90 MsgBox Err.Description
2692 91 Resume Err_btnLevel1_Click
2693 92
2694 93 End Sub
2695 94
2696 95 Private Sub btnLevel2_Click()
2697 96 On Error GoTo Err_btnLevel2_Click
2698 97
2699 98 Dim lngRecordNum As Long
2700 99 Dim strSQL As String
2701 100
2702 101 lngRecordNum = Me.CurrentRecord
2703 102
2704 103 DoCmd.Close
2705 104
2706 105 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2707 106 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2708 lngRecordNum
2709 107 Form!frmSchoolToPropLevelMaint!ChosenLevel = 2
2710 108
2711 109 'Update the selected property list
2712 110 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
2713 SCourseNumber_RK = " &
2714 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & " And
2715 AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2716 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2717 UNION SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK
2718 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & "
2719 And AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2720 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2721 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
2722 111
2723 112 Exit btnLevel2_Click
2724 113 Exit Sub
2725 114
2726 115 Err_btnLevel2_Click
2727 116 MsgBox Err.Description
2728 117 Resume Err_btnLevel2_Click
2729 118
2730 119 End Sub
2731 120
2732 121 Private Sub btnLevel3_Click()
2733 122 On Error GoTo Err_btnLevel3_Click
2734 123
2735 124 Dim lngRecordNum As Long
2736 125 Dim strSQL As String
2737 126
2738 127 lngRecordNum = Me.CurrentRecord
2739 128
2740 129 DoCmd.Close
2741 130
2742 131 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2743 132 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2744 lngRecordNum
2745 133 Form!frmSchoolToPropLevelMaint!ChosenLevel = 3
2746 134
2747 135 'Update the selected property list
2748 136 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
2749 SCourseNumber_RK = " &
2750 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & " And
2751 AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2752 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2753 UNION SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK
2754 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & "
2755 And AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2756 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2757 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
2758 137
2759 138 Exit btnLevel3_Click
2760 139 Exit Sub
2761 140
2762 141 Err_btnLevel3_Click
2763 142 MsgBox Err.Description
2764 143 Resume Err_btnLevel3_Click
2765 144
2766 145 End Sub
2767 146
2768 147 Private Sub btnLevel4_Click()
2769 148 On Error GoTo Err_btnLevel4_Click
2770 149
2771 150 Dim lngRecordNum As Long
2772 151 Dim strSQL As String
2773 152
2774 153 lngRecordNum = Me.CurrentRecord
2775 154
2776 155 DoCmd.Close
2777 156
2778 157 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2779 158 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2780 lngRecordNum
2781 159 Form!frmSchoolToPropLevelMaint!ChosenLevel = 4
2782 160
2783 161 'Update the selected property list
2784 162 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
2785 SCourseNumber_RK = " &
2786 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & " And
2787 AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2788 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2789 UNION SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK

```

```

2990 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & "
2991 And AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
2992 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
2993 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
2994 167
2995 168 Exit btnLevel4_Click
2996 169 Exit Sub
2997 170
2998 171 Err_btnLevel4_Click
2999 172 MsgBox Err.Description
3000 173 Resume Err_btnLevel4_Click
3001 174
3002 175 End Sub
3003 176
3004 177 Private Sub btnLevel5_Click()
3005 178 On Error GoTo Err_btnLevel5_Click
3006 179
3007 180 Dim lngRecordNum As Long
3008 181 Dim strSQL As String
3009 182
3010 183 lngRecordNum = Me.CurrentRecord
3011 184
3012 185 DoCmd.Close
3013 186
3014 187 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
3015 188 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
3016 lngRecordNum
3017 189 Form!frmSchoolToPropLevelMaint!ChosenLevel = 5
3018 190
3019 191 'Update the selected property list
3020 192 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
3021 SCourseNumber_RK = " &
3022 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & " And
3023 AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
3024 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
3025 UNION SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK
3026 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & "
3027 And AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
3028 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
3029 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
3030 193
3031 194 Exit btnLevel5_Click
3032 195 Exit Sub
3033 196
3034 197 Err_btnLevel5_Click
3035 198 MsgBox Err.Description
3036 199 Resume Err_btnLevel5_Click
3037 200
3038 201 End Sub
3039 202
3040 203 Private Sub btnLevel6_Click()
3041 204 On Error GoTo Err_btnLevel6_Click
3042 205
3043 206 Dim lngRecordNum As Long
3044 207 Dim strSQL As String
3045 208
3046 209 lngRecordNum = Me.CurrentRecord
3047 210
3048 211 DoCmd.Close
3049 212
3050 213 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
3051 214 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
3052 lngRecordNum
3053 215 Form!frmSchoolToPropLevelMaint!ChosenLevel = 6
3054 216
3055 217 'Update the selected property list
3056 218 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
3057 SCourseNumber_RK = " &
3058 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & " And
3059 AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
3060 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
3061 UNION SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK
3062 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & " " & "
3063 And AMOS_RK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & " " & "
3064 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & " " & "
3065 Form!frmSchoolToPropLevelMaint!lstLevel0.RowSource = strSQL
3066 219
3067 220 Exit btnLevel6_Click
3068 221 Exit Sub
3069 222
3070 223 Err_btnLevel6_Click
3071 224 MsgBox Err.Description
3072 225 Resume Err_btnLevel6_Click
3073 226
3074 227 End Sub
3075 228
3076 229 Private Sub cmbCourseNumberFind_AfterUpdate()
3077 230
3078 231 Dim R As Recordset
3079 232 Set R = Me.RecordsetClone
3080 233 R.FindFirst "SCourseNumber_PK = " & Chr(34) &
3081 Me!cmbCourseNumberFind & Chr(34)
3082 234
3083 235 Me.Bookmark = R.Bookmark
3084 236 Me!cmbCourseNumberFind = Null
3085 237 Me!SCourseNumber_PK.SetFocus
3086 238
3087 239 End Sub
3088 240
3089 241 Private Sub Form_Current()
3090 242
3091 243 Dim strSQL As String
3092 244 Dim lngLevel As Integer
3093 245
3094 246 'Update the Level 0 list
3095 247 lngLevel = 0
3096 248 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
3097 SCourseNumber_RK = " & Me!SCourseNumber_PK.Value & " " & " And AMOS_RK =
3098 " & Me!AMOS_PK.Value & " " & " And Level = " & lngLevel & " " & " UNION
3099 SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK = " &
3100 Me!SCourseNumber_PK.Value & " " & " And AMOS_RK = " & Me!AMOS_PK.Value
3101 & " " & " And Level = " & lngLevel & " " & "
3102 249 Me!lstLevel0.RowSource = strSQL
3103 250
3104 251 'Update the Level 1 list
3105 252 lngLevel = 1
3106 253 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
3107 SCourseNumber_RK = " & Me!SCourseNumber_PK.Value & " " & " And AMOS_RK =
3108 " & Me!AMOS_PK.Value & " " & " And Level = " & lngLevel & " " & " UNION
3109 SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK = " &
3110 Me!SCourseNumber_PK.Value & " " & " And AMOS_RK = " & Me!AMOS_PK.Value
3111 & " " & " And Level = " & lngLevel & " " & "
3112 254 Me!lstLevel1.RowSource = strSQL
3113 255
3114 256 'Update the Level 2 list
3115 257 lngLevel = 2
3116 258 strSQL = "SELECT (FPropertyName_RK) FROM FUND_SCH_PROP WHERE
3117 SCourseNumber_RK = " & Me!SCourseNumber_PK.Value & " " & " And AMOS_RK =
3118 " & Me!AMOS_PK.Value & " " & " And Level = " & lngLevel & " " & " UNION
3119 SELECT (LPropertyName_RK) FROM LOG_SCH_PROP WHERE SCourseNumber_RK = " &
3120 Me!SCourseNumber_PK.Value & " " & " And AMOS_RK = " & Me!AMOS_PK.Value

```

```

3120 & "" & " And Level = " & IntLevel & ""
3121 Me.lstLevel2.RowSource = strSQL
3122
3123
3124
3125 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3126 SCourseNumber_FK = " & Me.SCourseNumber_PK.Value & "" & " And AMOS_FK =
3127 "" & Me.AMOS_PK.Value & "" & " And Level = " & IntLevel & "" & " UNION
3128 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE SCourseNumber_FK = " &
3129 Me.SCourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3130 & "" & " And Level = " & IntLevel & ""
3131 Me.lstLevel3.RowSource = strSQL
3132
3133
3134
3135 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3136 SCourseNumber_FK = " & Me.SCourseNumber_PK.Value & "" & " And AMOS_FK =
3137 "" & Me.AMOS_PK.Value & "" & " And Level = " & IntLevel & "" & " UNION
3138 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE SCourseNumber_FK = " &
3139 Me.SCourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3140 & "" & " And Level = " & IntLevel & ""
3141 Me.lstLevel4.RowSource = strSQL
3142
3143
3144
3145 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3146 SCourseNumber_FK = " & Me.SCourseNumber_PK.Value & "" & " And AMOS_FK =
3147 "" & Me.AMOS_PK.Value & "" & " And Level = " & IntLevel & "" & " UNION
3148 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE SCourseNumber_FK = " &
3149 Me.SCourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3150 & "" & " And Level = " & IntLevel & ""
3151 Me.lstLevel5.RowSource = strSQL
3152
3153
3154
3155 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3156 SCourseNumber_FK = " & Me.SCourseNumber_PK.Value & "" & " And AMOS_FK =
3157 "" & Me.AMOS_PK.Value & "" & " And Level = " & IntLevel & "" & " UNION
3158 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE SCourseNumber_FK = " &
3159 Me.SCourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3160 & "" & " And Level = " & IntLevel & ""
3161 Me.lstLevel6.RowSource = strSQL
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249

```

```

3250 FROM qryUnionAllProperty LEFT JOIN qryUnionPropertyByLevel ON
3251 [qryUnionAllProperty].[PropertyName_FK] =
3252 [qryUnionPropertyByLevel].[PropertyName_FK] WHERE
3253 ([qryUnionPropertyByLevel].[PropertyName_FK] Is Null);
3254
3255 Me.lstLevelUnUsed.RowSource = strSQL
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379

```

```

3380 6 Option Explicit
3381 7
3382 8 Private Sub Close_Click()
3383 9 On Error GoTo Err_Close_Click
3384 10
3385 11 Dim stDocName As String
3386 12 Dim stLinkCriteria As String
3387 13
3388 14 ' Close current form
3389 15 DoCmd.Close
3390 16
3391 17 ' Open specified form
3392 18 stDocName = "frmPreprocessing&ExecutionSwitchboard"
3393 19 DoCmd.OpenForm stDocName, , stLinkCriteria
3394 20
3395 21 Exit_Close_Click:
3396 22 Exit Sub
3397 23
3398 24 Err_Close_Click:
3399 25 MsgBox Err.Description
3400 26 Resume Exit_Close_Click
3401 27
3402 28 End Sub
3403 29
3404 30 Private Sub cmbPefFind_AfterUpdate()
3405 31
3406 32 Dim R As Recordset
3407 33 Set R = Me.RecordsetClone
3408 34 R.FindFirst "[PEF] = " & Chr(34) & Me[cmbPefFind] & Chr(34)
3409 35 Me.Bookmark = R.Bookmark
3410 36 Me[cmbSSNFind] = Null
3411 37 Me.SSN_PK.SetFocus
3412 38
3413 39 End Sub
3414 40
3415 41 Private Sub cmbSSNFind_AfterUpdate()
3416 42
3417 43 Dim R As Recordset
3418 44 Set R = Me.RecordsetClone
3419 45 R.FindFirst "[SSN] = " & Chr(34) & Me[cmbSSNFind] & Chr(34)
3420 46 Me.Bookmark = R.Bookmark
3421 47 Me[cmbSSNFind] = Null
3422 48 Me.Close.SetFocus
3423 49
3424 50 End Sub
3425 51
3426 52 Private Sub Delete_Click()
3427 53 On Error GoTo Err_Delete_Click
3428 54
3429 55 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
3430 56 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
3431 57
3432 58 Exit_Delete_Click:
3433 59 Exit Sub
3434 60
3435 61 Err_Delete_Click:
3436 62 MsgBox Err.Description
3437 63 Resume Exit_Delete_Click
3438 64
3439 65 End Sub
3440 66
3441 67 Private Sub Form_Current()
3442 68 Dim stSQL As String
3443 69
3444 70 ' Update the PEF List
3445 71 stSQL = "SELECT DISTINCTROW [MARINE] [PEF] FROM MARINE LEFT JOIN PEF
3446 72 ON [MARINE].[PEF] = [PEF].[PEF_PK] WHERE ([PEF].[PEF_PK] Is Null) ORDER BY
3447 73 [MARINE].[PEF] RowSource = stSQL
3448 74
3449 75 End Sub
3450 76
3451 77 Form: frmSpecialAssignment
3452 78 Code
3453 79
3454 80 1 Attribute VB_Name = "Form_frmSpecialAssignment"
3455 81 2 Attribute VB_Creatable = True
3456 82 3 Attribute VB_PredeclaredId = True
3457 83 4 Attribute VB_Exposed = False
3458 84 5 Option Compare Database
3459 85 6 Option Explicit
3460 86
3461 87 Private Sub cmbClassNumber_AfterUpdate()
3462 88
3463 89 Dim stConvert As String
3464 90
3465 91 Me.tbReportDate = Me.cmbClassNumber.Column(2)
3466 92 Me.tbMCC = Me.cmbClassNumber.Column(4)
3467 93
3468 94 ' Convert the fiscal year to a two digit number for RD3 File
3469 95 stConvert = CSng(Me.cmbClassNumber.Column(3))
3470 96 stConvert = Right(stConvert, 2)
3471 97 Me.tbFY = CInt(stConvert)
3472 98
3473 99 Me.tbAssignmentType = "S"
3474 100
3475 101 End Sub
3476 102
3477 103 Private Sub cmbClassNumber_Enter()
3478 104
3479 105 Dim stSQL As String
3480 106
3481 107 ' This query finds the class numbers, report dates and class
3482 108 numbering dates associated with the chosen school
3483 109 stSQL = "SELECT BNA_EXTRACT.ClassNumber, BNA_EXTRACT.ReportDate,
3484 110 BNA_EXTRACT.FiscalYear_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT INNER JOIN
3485 111 SCH_TGT_MOS ON (BNA_EXTRACT.ClassNumber_PK =
3486 112 SCH_TGT_MOS.TargetMOS_FK) AND (BNA_EXTRACT.TargetMOS_PK =
3487 113 SCH_TGT_MOS.TargetMOS_FK) WHERE BNA_EXTRACT.CourseNumber_PK =
3488 114 Form!frmSpecialAssignment!cmbCourseNumber AND SCH_TGT_MOS.AMOS_FK =
3489 115 Form!frmSpecialAssignment!cmbAMOS"
3490 116
3491 117 Me.cmbClassNumber.RowSource = stSQL
3492 118
3493 119 End Sub
3494 120
3495 121 Private Sub cmbCourseNumber_AfterUpdate()
3496 122
3497 123 Me.tbAMOS = Me.cmbCourseNumber.Column(1)
3498 124
3499 125 End Sub
3500 126
3501 127 Private Sub cmbSSN_AfterUpdate()
3502 128
3503 129 Me.tbSSN = Me.cmbSSN.Column(1)
3504 130 Me.tbGradDate = Me.cmbSSN.Column(2)
3505 131
3506 132 End Sub
3507 133
3508 134 Private Sub cmbSSN_Enter()
3509 135
3510 136 Dim stSQL As String
3511 137
3512 138 stSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SSN,
3513 139 MARINE.GradDate FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
3514 140 ASSIGNMENT.SSN_FK WHERE ((ASSIGNMENT.SSN_FK Is Null));"
3515 141
3516 142 Me.cmbSSN.RowSource = stSQL
3517 143
3518 144 End Sub
3519 145 Private Sub cmbSSNFind_AfterUpdate()
3520 146
3521 147 Dim R As Recordset
3522 148 Set R = Me.RecordsetClone
3523 149 R.FindFirst "[SSN] = " & Chr(34) & Me[cmbSSNFind] & Chr(34)
3524 150 Me.Bookmark = R.Bookmark
3525 151 Me[cmbSSNFind] = Null
3526 152 Me.Close.SetFocus
3527 153
3528 154 End Sub
3529 155
3530 156 Private Sub Delete_Click()
3531 157 On Error GoTo Err_Delete_Click
3532 158
3533 159 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
3534 160 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
3535 161
3536 162 Exit_Delete_Click:
3537 163 Exit Sub
3538 164
3539 165 Err_Delete_Click:
3540 166 MsgBox Err.Description
3541 167 Resume Exit_Delete_Click
3542 168
3543 169 End Sub
3544 170
3545 171 Private Sub Close_Click()
3546 172 On Error GoTo Err_Close_Click
3547 173
3548 174 Dim stDocName As String
3549 175 Dim stLinkCriteria As String
3550 176
3551 177 ' Close current form
3552 178 DoCmd.Close
3553 179
3554 180 ' Open specified form
3555 181 stDocName = "frmPreprocessing&ExecutionSwitchboard"
3556 182 DoCmd.OpenForm stDocName, , stLinkCriteria
3557 183
3558 184 Exit_Close_Click:
3559 185 Exit Sub
3560 186
3561 187 Err_Close_Click:
3562 188 MsgBox Err.Description
3563 189 Resume Exit_Close_Click
3564 190
3565 191 End Sub
3566 192
3567 193 Private Sub Form_Current()
3568 194
3569 195 ' Prevents user from accidentally making special assignments to
3570 196 marines assigned normally
3571 197
3572 198 If IsNull(Me.cmbSSN.Value) = False Then
3573 199 Me.cmbSSN.Enabled = False
3574 200 Me.cmbCourseNumber.Enabled = False
3575 201 Me.cmbClassNumber.Enabled = False
3576 202
3577 203 ' Allows the user to enter values for the fundamental equation
3578 204 Else
3579 205 Me.cmbSSN.Enabled = True
3580 206 Me.cmbCourseNumber.Enabled = True
3581 207 Me.cmbClassNumber.Enabled = True
3582 208
3583 209 End If
3584 210
3585 211 End Sub
3586 212
3587 213 Private Sub Form_Open(Cancel As Integer)
3588 214
3589 215 DoCmd.GoToRecord acDataForm, "frmSpecialAssignment", acNewRec
3590 216
3591 217 End Sub
3592 218
3593 219 Form: frmSpecialAssignmentOfUnassignedMarine
3594 220 Code
3595 221
3596 222 1 Attribute VB_Name = "Form_frmSpecialAssignmentOfUnassignedMarine"
3597 223 2 Attribute VB_Creatable = True
3598 224 3 Attribute VB_PredeclaredId = True
3599 225 4 Attribute VB_Exposed = False
3600 226 5 Option Compare Database
3601 227 6 Option Explicit
3602 228
3603 229 Private Sub cmbClassNumber_AfterUpdate()
3604 230
3605 231 Dim stConvert As String
3606 232 Dim stSSN As String
3607 233 Dim R As Recordset
3608 234
3609 235 Me.tbReportDate = Me.cmbClassNumber.Column(2)
3610 236 Me.tbMCC = Me.cmbClassNumber.Column(4)
3611 237
3612 238 ' Convert the fiscal year to a two digit number for RD3 File
3613 239 stConvert = CSng(Me.cmbClassNumber.Column(3))
3614 240 stConvert = Right(stConvert, 2)
3615 241 Me.tbFY = CInt(stConvert)
3616 242
3617 243 Me.tbAssignmentType = "S"
3618 244
3619 245 ' Keep track of the current SSN
3620 246 stSSN = Me.cmbSSN
3621 247
3622 248 Me.tbAMOS = Me.cmbCourseNumber.Column(1)
3623 249
3624 250 ' Updates the current form and subform
3625 251 Me.Requery
3626 252 Me.Repaint
3627 253
3628 254 ' Moves the record back to the current SSN
3629 255 Set R = Me.RecordsetClone
3630 256 R.FindFirst "[SSN] = " & Chr(34) & stSSN & Chr(34)
3631 257 Me.Bookmark = R.Bookmark
3632 258
3633 259 End Sub
3634 260
3635 261 Private Sub cmbClassNumber_Enter()
3636 262
3637 263 Dim stSQL As String
3638 264
3639 265 stSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SSN,
3640 266 MARINE.GradDate FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
3641 267 ASSIGNMENT.SSN_FK WHERE ((ASSIGNMENT.SSN_FK Is Null));"
3642 268
3643 269 Me.cmbSSN.RowSource = stSQL
3644 270
3645 271 End Sub

```

```

3640 44 ' This query finds the class numbers, report dates and class
3641 converting dates associated with the chosen school
3642 45 strSQL = "SELECT BNA_EXTRACT.ClassNumber_PK,
3643 BNA_EXTRACT.ClassConvDate, BNA_EXTRACT.ReportDate,
3644 BNA_EXTRACT.FiscalYear_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT INNER JOIN
3645 SCH_TGT_MOS ON (BNA_EXTRACT.CourseNumber_PK =
3646 SCH_TGT_MOS.CourseNumber_FK) AND (BNA_EXTRACT.TargetMOS_PK =
3647 SCH_TGT_MOS.TargetMOS_FK) WHERE BNA_EXTRACT.CourseNumber_PK =
3648 Form!frmSpecialAssignmentOfUnassignedMarine!cmbCourseNumber AND
3649 SCH_TGT_MOS.AMOS_FK =
3650 Me.cmbClassNumber.RowSource = strSQL
3651 47
3652 48 End Sub
3653 49
3654 50 Private Sub cmbCourseNumber_AfterUpdate()
3655 51
3656 52 Dim strSSN As String
3657 53 Dim R As Recordset
3658 54
3659 55 strSSN = Me.cmbSSN
3660 56
3661 57 ' Keep track of the current SSN
3662 58 strSSN = Me.cmbSSN
3663 59
3664 60 Me.tblAMOS = Me!cmbCourseNumber.Column(1)
3665 61
3666 62 ' Updates the current form and subform
3667 63 Me.Requery
3668 64 Me.Repaint
3669 65
3670 66 ' Moves the record back to the current SSN
3671 67 Set R = Me.RecordsetClone
3672 68 R.FindFirst "[SSN_FK] = " & Chr(34) & strSSN & Chr(34)
3673 69 Me.Bookmark = R.Bookmark
3674 70
3675 71 End Sub
3676 72
3677 73 Private Sub cmbSSN_AfterUpdate()
3678 74
3679 75 Me.tblSQL = Me!cmbSSN.Column(1)
3680 76 Me.tblGradDate = Me!cmbSSN.Column(2)
3681 77
3682 78 End Sub
3683 79
3684 80 Private Sub cmbSSN_Enter()
3685 81
3686 82 Dim strSQL As String
3687 83
3688 84 strSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SCH,
3689 MARINE.GradDate FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
3690 ASSIGNMENT.SSN_FK WHERE ((ASSIGNMENT.SSN_FK) Is Null);"
3691 85 Me.cmbSSN.RowSource = strSQL
3692 86
3693 87 End Sub
3694 88
3695 89 Private Sub cmbSSNFind_AfterUpdate()
3696 90
3697 91 Dim R As Recordset
3698 92 Set R = Me.RecordsetClone
3699 93 R.FindFirst "[SSN_FK] = " & Chr(34) & Me!cmbSSNFind & Chr(34)
3700 94 Me.Bookmark = R.Bookmark
3701 95 Me!cmbSSNFind = Null
3702 96 Me.CloseSelfFocus
3703 97
3704 98 End Sub
3705 99
3706 100 Private Sub Delete_Click()
3707 101 On Error GoTo Err_Delete_Click
3708 102
3709 103
3710 104 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 8, acMenuVer70
3711 105 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 6, acMenuVer70
3712 106
3713 107 Err_Delete_Click:
3714 108 Exit Sub
3715 109
3716 110 Err_Delete_Click:
3717 111 MsgBox Err.Description
3718 112 Resume Err_Delete_Click
3719 113
3720 114 End Sub
3721 115
3722 116 Private Sub Close_Click()
3723 117 On Error GoTo Err_Close_Click
3724 118
3725 119 ' Close current form
3726 120 DoCmd.Close
3727 121
3728 122 Err_Close_Click:
3729 123 Exit Sub
3730 124
3731 125 Err_Close_Click:
3732 126 MsgBox Err.Description
3733 127 Resume Err_Close_Click
3734 128
3735 129 End Sub
3736 130
3737 131 Private Sub Form_Current()
3738 132
3739 133 Me!cmbSSN.Enabled = True
3740 134 Me!cmbCourseNumber.Enabled = True
3741 135 Me!cmbClassNumber.Enabled = True
3742 136
3743 137
3744 138
3745 139 End Sub
3746 140
3747 141 Private Sub Form_Open(Cancel As Integer)
3748 142
3749 143
3750 144 DoCmd.GoToRecord acDataForm,
3751 frmSpecialAssignmentOfUnassignedMarine, acNewRec
3752 145
3753 146
3754 147 End Sub
3755 148
3756 Form: frmUnassignedMarines
3757 Code
3758 1 Attribute VB_Name = "Form_frmUnassignedMarines"
3759 2 Attribute VB_Creatable = True
3760 3 Attribute VB_PredeclaredId = True
3761 4 Attribute VB_Exposed = False
3762 5 Option Compare Database
3763 6 Option Explicit
3764 7
3765 8 Private Sub Close_Click()
3766 9 On Error GoTo Err_Close_Click
3767 10
3768 11 Dim sDocName As String
3769 12 Dim sLinkCriteria As String

```

```

3770 13
3771 14 ' Close the current form
3772 15 DoCmd.Close
3773 16
3774 17 ' Open specified form
3775 18 sDocName = "frmAnalyzeResult"
3776 19 DoCmd.OpenForm sDocName, , , sLinkCriteria
3777 20 Exit_Close_Click
3778 21 Exit Sub
3779 22
3780 23 Err_Close_Click:
3781 24 MsgBox Err.Description
3782 25 Resume Exit_Close_Click
3783 26
3784 27 End Sub
3785 28
3786 29 Private Sub cmbSSNFind_AfterUpdate()
3787 30
3788 31 Dim R As Recordset
3789 32 Set R = Me.RecordsetClone
3790 33 R.FindFirst "[SSN_PK] = " & Chr(34) & Me!cmbSSNFind & Chr(34)
3791 34 Me.Bookmark = R.Bookmark
3792 35 Me!cmbSSNFind = Null
3793 36 Me.CloseSelfFocus
3794 37
3795 38 End Sub
3796 39
3797 40 Private Sub Form_Current()
3798 41
3799 42 Dim strSQL As String
3800 43 Dim rec As Recordset
3801 44 Dim db1 As Database
3802 45
3803 46 Set db1 = CurrentDb()
3804 47
3805 48 ' Calculates the number of unassigned marines.
3806 49 strSQL = "SELECT Count(PEF) AS TotalUnassigned FROM
3807 50 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
3808 51 Me.txtTotalUnassigned = rec.TotalUnassigned
3809 52
3810 53 End Sub
3811 54 Private Sub btnAssign_Click()
3812 55 On Error GoTo Err_btnAssign_Click
3813 56
3814 57 Dim sDocName As String
3815 58
3816 59 sDocName = "frmSpecialAssignmentOfUnassignedMarine"
3817 60
3818 61 DoCmd.OpenForm sDocName
3819 62
3820 63 Err_btnAssign_Click:
3821 64 Exit Sub
3822 65
3823 66 Err_btnAssign_Click:
3824 67 MsgBox Err.Description
3825 68 Resume Err_btnAssign_Click
3826 69
3827 70 End Sub
3828 71
3829 Form: subfrmFundamentalsProperty
3830 Code
3831 1 Attribute VB_Name = "Form_subfrmFundamentalsProperty"
3832 2 Attribute VB_Creatable = True
3833 3 Attribute VB_PredeclaredId = True
3834 4 Attribute VB_Exposed = False
3835 5 Option Compare Database
3836 6 Option Explicit
3837 7 Private Sub cmbValue_FK_AfterUpdate()
3838 8
3839 9 Form!frmFundamentalsProperty!FFPropertyName_PK.SetFocus
3840 10
3841 11 End Sub
3842 12
3843 13 Private Sub cmbValue_FK_BeforeUpdate(Cancel As Integer)
3844 14
3845 15 On Error GoTo Err_cmbValue_FK_BeforeUpdate
3846 16
3847 17 Dim db As Database
3848 18 Dim rec As Recordset
3849 19 Dim strSQL As String
3850 20 Dim Convert As Variant
3851 21
3852 22 ' Places value in a form agreeable to Access (e.g. "1" or "10")
3853 23 Convert = "" &
3854 24 Form!frmFundamentalsProperty!subfrmFundamentalsProperty!cmbValue_FK & ""
3855 25
3856 26 ' Opens the table PROPERTY_VALUE
3857 27 strSQL = "PROPERTY_VALUE"
3858 28 Set rec = CurrentDb()
3859 29 Set rec = db.OpenRecordset(strSQL, dbOpenSnapshot)
3860 30
3861 31 ' Searches for the value entered by the user in the
3862 32 rec.FindFirst "[Value_FK] = " & Convert
3863 33
3864 34 ' If the value is not in the PROPERTY_VALUE table, it is added.
3865 35 If rec.NoMatch = True Then
3866 36
3867 37 ' Disables the action query confirmation message
3868 38 Application.SetOption "Confirm Action Queries", False
3869 39
3870 40 ' Enters the value into the PROPERTY_VALUE table
3871 41 DoCmd.OpenQuery "qryUpdateProperty_Value"
3872 42
3873 43 ' Updates the drop down list for the combobox
3874 44 strSQL = "SELECT DISTINCT " & " FROM MARINE;"
3875 45 Form!frmFundamentalsProperty!MarineField & " FROM MARINE;"
3876 46
3877 47 Form!frmFundamentalsProperty!subfrmFundamentalsProperty!cmbValue_FK.RowSou
3878 48 rce = strSQL
3879 49
3880 50 ' Enables the action query confirmation message
3881 51 Application.SetOption "Confirm Action Queries", True
3882 52 End If
3883 53
3884 54 rec.Close
3885 55
3886 56 Err_cmbValue_FK_BeforeUpdate:
3887 57 Exit Sub
3888 58
3889 59 Err_cmbValue_FK_BeforeUpdate:
3890 60 MsgBox Err.Description
3891 61 Application.SetOption "Confirm Action Queries", True
3892 62 Resume Err_cmbValue_FK_BeforeUpdate
3893 63
3894 64
3895 65 End Sub
3896 66
3897 Form: subfrmFundamentalsPropertyList
3898 Code
3899 1 Attribute VB_Name = "Form_subfrmFundamentalsPropertyList"

```



```

3900 2 Attribute VB_Creatable = True
3901 3 Attribute VB_PredeclaredId = True
3902 4 Attribute VB_Exposed = False
3903 5 Option Compare Database
3904 6 Option Explicit
3905 7
3906 8 Private Sub cmbValue_FK_BeforeUpdate(Cancel As Integer)
3907 9 On Error GoTo Err_cmbValue_FK_BeforeUpdate
3908 10
3909 11 Dim db As Database
3910 12 Dim rec As Recordset
3911 13 Dim strSQL As String
3912 14 Dim Convert As Variant
3913 15
3914 16 'Opens the subfrmFundamentalsProperty so the following code will
3915 17 DoCmd.OpenForm "subfrmFundamentalsProperty", , , , acHidden
3916 18
3917 19
3918 20 'Places value in a form agreeable to Access (e.g. 'HT' or '10')
3919 21 Convert = "" & Form([subfrmFundamentalsProperty]/cmbValue_FK &
3920 22 'Opens the table PROPERTY_VALUE
3921 23 strSQL = "PROPERTY_VALUE"
3922 24
3923 25 Set db = CurrentDb()
3924 26 Set rec = db.OpenRecordset(strSQL, dbOpenSnapshot)
3925 27
3926 28 rec.FindFirst "Value_PK = " & Convert
3927 29 If rec.NoMatch = True Then
3928 30 'Disables the action query confirmation message
3929 31 Application.SetOption "Confirm Action Queries", False
3930 32
3931 33 DoCmd.OpenQuery "qryUpdateProperty_Value"
3932 34 strSQL = "SELECT DISTINCT ' &
3933 35 Format(IIf(FundamentalsProperty/MarineField & ' FROM MARINE;"
3934 36 Me.cmbValue_FK.RowSource = strSQL
3935 37
3936 37 'Enables the action query confirmation message
3937 38 Application.SetOption "Confirm Action Queries", True
3938 39 End If
3939 40
3940 41 rec.Close
3941 42
3942 43 'Closes the subfrmFundamentalsProperty without saving changes to
3943 44 DoCmd.Close acForm, "subfrmFundamentalsProperty"
3944 45
3945 46
3946 47 Exit_cmbValue_FK_BeforeUpdate:
3947 48 Exit Sub
3948 49
3949 50 Err_cmbValue_FK_BeforeUpdate:
3950 51 MsgBox Err.Description
3951 52 Application.SetOption "Confirm Action Queries", True
3952 53 Resume Exit_cmbValue_FK_BeforeUpdate
3953 54
3954 55 End Sub
3955 56
3956 57 Private Sub Value_FK_BeforeUpdate(Cancel As Integer)
3957 58 On Error GoTo Err_Value_FK_BeforeUpdate
3958 59
3959 60 Dim db As Database
3960 61 Dim rec As Recordset
3961 62 Dim strSQL As String
3962 63 Dim Convert As Variant
3963 64
3964 65 'Places value in a form agreeable to Access (e.g. 'HT' or '10')
3965 66 Convert = "" &
3966 67 Format(IIf(FundamentalsProperty/subfrmFundamentalsPropertyListValue_FK & ""
3967 68 'Opens the table PROPERTY_VALUE
3968 69 strSQL = "PROPERTY_VALUE"
3969 70
3970 71 Set db = CurrentDb()
3971 72 Set rec = db.OpenRecordset(strSQL, dbOpenSnapshot)
3972 73
3973 74 rec.FindFirst "Value_PK = " & Convert
3974 75 'This If Then statement checks to see if the entered value is in
3975 76 'PROPERTY_VALUE table. If not, it is added to the table.
3976 77 If rec.NoMatch = True Then
3977 78 'Disables the action query confirmation message
3978 79 Application.SetOption "Confirm Action Queries", False
3979 80
3980 81 DoCmd.OpenQuery "qryUpdateProperty_ValueList"
3981 82
3982 83 'Enables the action query confirmation message
3983 84 Application.SetOption "Confirm Action Queries", True
3984 85 End If
3985 86
3986 87 rec.Close
3987 88
3988 89 Exit_Value_FK_BeforeUpdate:
3989 90 Exit Sub
3990 91
3991 92 Err_Value_FK_BeforeUpdate:
3992 93 MsgBox Err.Description
3993 94 Application.SetOption "Confirm Action Queries", True
3994 95 Resume Exit_Value_FK_BeforeUpdate
3995 96
3996 97 End Sub
3997 98
3998 99
4000 100 Module: modFitnessDetermination
4001 101 Code
4002 102 1 Attribute VB_Name = "modFitnessDetermination"
4003 103 2 Option Compare Database
4004 104 3 Option Explicit
4005 105 4 Dim lngStart, lngEnd As Long
4006 106 5
4007 107 6
4008 108 7 Public Function FundPropTest(FundPropName As String, SSN As String) As
4009 109
4010 110 Dim rec As Recordset, rec2 As Recordset, rec3 As Recordset
4011 111 Dim strMarField As String, strFundOperator As String, strSQL As
4012 112 Dim strMarValueSQL As String, strFundValueSQL As String, strInput As
4013 113 Dim verFundValue, verMarValue
4014 114 Dim db1 As Database
4015 115
4016 116 FundPropName = "" & FundPropName & "" 'This expression
4017 117 properly formulates the argument FundPropName, by placing a " in front of
4018 118 and behind the value found in FundPropName.
4019 119
4020 120 Set db1 = CurrentDb()
4021 121
4022 122 strSQL = "SELECT MarineField, Operator FROM FUNDAMENTAL_PROPERTY
4023 123 WHERE SSN_PK = " & SSN & " & FundPropName
4024 124 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
4025 125 strMarField = rec1.MarineField
4026 126
4027 127 strFundOperator = rec1.Operator
4028 128
4029 129
4030 130 strInput = "" & SSN & ""
4031 131 strMarValueSQL = "SELECT ' & strMarField & ' AS [Value] FROM MARINE
4032 132 WHERE SSN_PK = " & strInput
4033 133 Set rec2 = db1.OpenRecordset(strMarValueSQL, dbOpenSnapshot)
4034 134 verMarValue = rec2.Value
4035 135
4036 136 strFundValueSQL = "SELECT Value_FK FROM FUND_PROP_VAL WHERE
4037 137 FPropertyName_PK = " & FundPropName
4038 138 Set rec3 = db1.OpenRecordset(strFundValueSQL, dbOpenSnapshot)
4039 139 verFundValue = rec3.Value_FK
4040 140
4041 141 'Convert verFundValue and verMarValue to a numeric, if possible
4042 142 If IsNumeric(verFundValue) And IsNumeric(verMarValue) Then
4043 143 verFundValue = CInt(verFundValue)
4044 144 verMarValue = CInt(verMarValue)
4045 145 Else
4046 146 verFundValue = CStr(verFundValue)
4047 147 verMarValue = CStr(verMarValue)
4048 148 End If
4049 149
4050 150
4051 151 Select Case strFundOperator
4052 152 Case "="
4053 153 If verMarValue = verFundValue Then
4054 154 FundPropTest = True
4055 155 End If
4056 156 Case "<"
4057 157 If verMarValue < verFundValue Then
4058 158 FundPropTest = True
4059 159 End If
4060 160 Case "<="
4061 161 If verMarValue <= verFundValue Then
4062 162 FundPropTest = True
4063 163 End If
4064 164 Case ">"
4065 165 If verMarValue > verFundValue Then
4066 166 FundPropTest = True
4067 167 End If
4068 168 Case ">="
4069 169 If verMarValue >= verFundValue Then
4070 170 FundPropTest = True
4071 171 End If
4072 172 Case "<>"
4073 173 If verMarValue <> verFundValue Then
4074 174 FundPropTest = True
4075 175 End If
4076 176 Case "<>="
4077 177 If verMarValue <>= verFundValue Then
4078 178 FundPropTest = True
4079 179 End If
4080 180 Case "><"
4081 181 If verMarValue > verFundValue And IsNumeric(verMarValue)
4082 182 verFundValue = CInt(verFundValue)
4083 183 verMarValue = CInt(verMarValue)
4084 184 Else
4085 185 verFundValue = CStr(verFundValue)
4086 186 verMarValue = CStr(verMarValue)
4087 187 End If
4088 188 If verMarValue = verFundValue Then
4089 189 FundPropTest = True
4090 190 End If
4091 191 rec3.MoveNext
4092 192 Loop
4093 193 End Select
4094 194 rec2.Close
4095 195 rec3.Close
4096 196 db1.Close
4097 197
4098 198 End Function
4099 199 Public Function LogOrTest(Input1 As Boolean, Input2 As Boolean) As
4100 200 Boolean
4101 201 If Input1 Or Input2 Then
4102 202 LogOrTest = True
4103 203 Else
4104 204 LogOrTest = False
4105 205 End If
4106 206 End Function
4107 207 Public Function LogAndTest(Input1 As Boolean, Input2 As Boolean) As
4108 208 Boolean
4109 209 If Input1 And Input2 Then
4110 210 LogAndTest = True
4111 211 Else
4112 212 LogAndTest = False
4113 213 End If
4114 214
4115 215 End Function
4116 216 Public Function LogPropTest(LogPropName As String, SSN As String) As
4117 217 Boolean
4118 218 Dim lngNewBound As Integer, j As Integer, k As Integer, l As
4119 219 Integer, m As Integer
4120 220 Dim booOrOperator As Boolean, booAndOperator As Boolean,
4121 221 booNoOperator As Boolean
4122 222 Dim strLogicalEquation As String, strSQL As String, strInput As
4123 223 String
4124 224 Dim strWorkingEqn() As String
4125 225 Dim lngSpacePos As Integer 'position of the space
4126 226 Dim db1 As Database
4127 227 Dim rec As Recordset
4128 228 Set db1 = CurrentDb()
4129 229
4130 230 'Get the logical equation from the LOGICAL_PROPERTY table
4131 231 strInput = "" & LogPropName & ""
4132 232 strSQL = "SELECT LogicalEquation FROM LOGICAL_PROPERTY WHERE
4133 233 LPropertyName_PK = " & strInput & ""
4134 234 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
4135 235 strLogicalEquation = rec1.LogicalEquation
4136 236
4137 237 'Parse the string into an array
4138 238 'declare the temporary string as an array
4139 239 ReDim strWorkingEqn(0) As String
4140 240
4141 241 'Find the position of the first space
4142 242 lngSpacePos = InStr(strLogicalEquation, " ")
4143 243 'loop until no more spaces are found
4144 244 'InStr returns 0 if the space is not found
4145 245 Do Until lngSpacePos = 0
4146 246 'redimension the array, adding another element
4147 247 'UBound tells us how many elements there already are in the
4148 248 ReDim Preserve strWorkingEqn(UBound(strWorkingEqn) + 1)
4149 249 'set the new element to the characters from the beginning of the
4150 250 'string up to the first space. Left returns a number of
4151 251 'characters from the beginning of a string
4152 252 strWorkingEqn(UBound(strWorkingEqn)) =
4153 253 Trim(Left(strLogicalEquation, lngSpacePos - 1))
4154 254 lngSpacePos = InStr(strWorkingEqn, " ")
4155 255 'now we've copied these characters into the name, they can be
4156 256
4157 257 'removed from the original string. Right returns a number of
4158 258 'characters from the end of the string. It is the rightmost N
4159 259

```

```

4160 145 'characters, therefore we subtract the position of the space
4161 146 'the length of the string. This gives us the number of
4162 characters remaining
4163 147 strLogicalEquation = LTrim(RTrim(strLogicalEquation,
4164 Len(strLogicalEquation) - intSpacePos))
4165 148
4166 149 'Find the next space
4167 150 intSpacePos = InStr(strLogicalEquation, " ")
4168 151 Loop
4169 152
4170 153 For j = 1 To (Ubound(strWorkingEqn))
4171 154 If strWorkingEqn(j) = "(" Then
4172 155
4173 156 Line1:
4174 157 For k = (j + 1) To (Ubound(strWorkingEqn))
4175 158 Select Case strWorkingEqn(k)
4176 159 Case ")"
4177 160 For i = (j + 1) To (k - 1)
4178 161 Select Case strWorkingEqn(i)
4179 162 Case "O"
4180 163 boolOperator = True
4181 164 Case "A"
4182 165 boolAndOperator = True
4183 166 Case "N"
4184 167 boolNotOperator = True
4185 168 Case "T"
4186 169 If strWorkingEqn(j) = "(" Then
4187 170 If boolOperator Then
4188 171 strWorkingEqn(j) = "False"
4189 172 boolOperator = False
4190 173 Else
4191 174 strWorkingEqn(j) = "True"
4192 175 End If
4193 176 Else 'Executed after the first
4194 177 bracketed property is tested and stored in the location of "T"
4195 178 If boolOperator Then
4196 179 strWorkingEqn(i) = "False"
4197 180 boolOperator = False
4198 181 End If
4199 182 If boolOperator Then
4200 183 strWorkingEqn(j) =
4201 LogOrTest(CBool(strWorkingEqn(j)), CBool(strWorkingEqn(i)))
4202 184
4203 185 boolOperator = False
4204 186 ElseIf boolAndOperator Then
4205 187 strWorkingEqn(j) =
4206 LogAndTest(CBool(strWorkingEqn(j)), CBool(strWorkingEqn(i)))
4207 188
4208 189 boolOperator = False
4209 190 End If
4210 191 Case "F"
4211 192 If strWorkingEqn(j) = "(" Then
4212 193 If boolOperator Then
4213 194 strWorkingEqn(j) = "True"
4214 195 boolOperator = True
4215 196 Else
4216 197 strWorkingEqn(j) = "False"
4217 198 boolOperator = False
4218 199 End If
4219 200 Else 'Executed after the first
4220 201 bracketed property is tested and stored in the location of "T"
4221 202 If boolOperator Then
4222 203 strWorkingEqn(i) = "True"
4223 204 boolOperator = True
4224 205 End If
4225 206 If boolOperator Then
4226 207 strWorkingEqn(j) =
4227 LogOrTest(CBool(strWorkingEqn(j)), CBool(strWorkingEqn(i)))
4228 208
4229 209 boolOperator = False
4230 210 ElseIf boolAndOperator Then
4231 211 strWorkingEqn(j) =
4232 LogAndTest(CBool(strWorkingEqn(j)), CBool(strWorkingEqn(i)))
4233 212
4234 213 boolOperator = False
4235 214 End If
4236 215 Case "I" 'Do nothing
4237 216 Case Else
4238 217 If strWorkingEqn(j) = "(" Then
4239 218 If boolOperator Then
4240 219 strWorkingEqn(j) = Not
4241 strWorkingEqn(j)
4242 220 boolOperator = False
4243 221 Else
4244 222 strWorkingEqn(j) =
4245 FundPropTest(strWorkingEqn(j), SSN)
4246 223 boolOperator = True
4247 224 End If
4248 225 Else 'Executed after the first
4249 226 bracketed property is tested and stored in the location of "T"
4250 227 If boolOperator Then
4251 228 strWorkingEqn(i) = Not
4252 strWorkingEqn(i)
4253 229 boolOperator = False
4254 230 End If
4255 231 If boolOperator Then
4256 232 strWorkingEqn(j) =
4257 FundPropTest(strWorkingEqn(j), SSN)
4258 233 boolOperator = True
4259 234 End If
4260 235 Else
4261 236 strWorkingEqn(j) =
4262 FundPropTest(strWorkingEqn(j), SSN)
4263 237 boolOperator = True
4264 238 End If
4265 239 Next i
4266 240 Next k
4267 241 'Must transform the working equation to account
4268 242 for the bracketed expression evaluation.
4269 243
4270 244 intNewUBound = Ubound(strWorkingEqn) - (k - j)
4271 245 If intNewUBound < 1 Then
4272 246 ReDim strTransformedEqn(1 To intNewUBound)
4273 247 For m = 1 To j
4274 248 strTransformedEqn(m) = strWorkingEqn(m)
4275 249 Next m
4276 250 For m = (k + 1) To Ubound(strWorkingEqn)
4277 251 strTransformedEqn(m - k + j) =
4278 strWorkingEqn(m)
4279 252 Next m
4280 253 ReDim strWorkingEqn(1 To intNewUBound)
4281 254 For m = 1 To intNewUBound
4282 255 strWorkingEqn(m) = strTransformedEqn(m)
4283 256 Next m
4284 257 j = 1 'Set back to a value of 1.
4285 258 GoTo Line1 'Jump back to For j = 1 to
4286 259 (Ubound(strWorkingEqn)) loop
4287 260 Else
4288 261 LogPropTest = strWorkingEqn(j)
4289 262 Erase strWorkingEqn
4290 263 Erase strTransformedEqn

```

```

4290 264 Exit Function
4291 265 End If
4292 266 Case "I"
4293 267 j = k
4294 268 GoTo Line1 'Jump back to For j = 1 to
4295 269 (Ubound(strWorkingEqn)) loop
4296 270 End Select
4297 271 Next k
4298 272 End If
4299 273 Next j
4300 274 End If
4301 275 Next j
4302 276 End Function
4303 277 End Function
4304 278
4305 279 Public Sub Finesse()
4306 280 Dim strSQL As String, strClassName As String, strAMOS As String
4307 281 Dim strInput1 As String, strInput2 As String
4308 282 Dim i As Integer, j As Integer, k As Integer, l As Integer
4309 283 Dim intNormalFactor As Integer, intFit As Integer, intInput As
4310 284 Integer
4311 285 Dim intLevel1 As Integer, intLevel2 As Integer, intLevel3 As Integer
4312 286 Dim intLevel4 As Integer, intLevel5 As Integer, intLevel6 As Integer
4313 287 Dim angDesiredHeight As Single, angExplosion As Single, angSumOfFit As
4314 288 Single
4315 289 Dim angMandFitValue As Single
4316 290 Dim intGradDate As Long
4317 291 Dim db1 As Database
4318 292 Dim recClass As Recordset, recMarine As Recordset, recMarChFit As
4319 293 Recordset
4320 294 Dim recPEF As Recordset, recDesired As Recordset, recLDesired As
4321 295 Recordset
4322 296 Dim recMandProp As Recordset, recLDesired As Recordset
4323 297 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4324 298 Recordset
4325 299 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4326 300 Recordset
4327 301 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4328 302 Recordset
4329 303 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4330 304 Recordset
4331 305 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4332 306 Recordset
4333 307 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4334 308 Recordset
4335 309 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4336 310 Recordset
4337 311 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4338 312 Recordset
4339 313 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4340 314 Recordset
4341 315 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4342 316 Recordset
4343 317 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4344 318 Recordset
4345 319 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4346 320 Recordset
4347 321 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4348 322 Recordset
4349 323 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4350 324 Recordset
4351 325 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4352 326 Recordset
4353 327 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4354 328 Recordset
4355 329 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4356 330 Recordset
4357 331 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4358 332 Recordset
4359 333 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4360 334 Recordset
4361 335 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4362 336 Recordset
4363 337 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4364 338 Recordset
4365 339 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4366 340 Recordset
4367 341 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4368 342 Recordset
4369 343 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4370 344 Recordset
4371 345 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4372 346 Recordset
4373 347 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4374 348 Recordset
4375 349 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4376 350 Recordset
4377 351 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4378 352 Recordset
4379 353 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4380 354 Recordset
4381 355 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4382 356 Recordset
4383 357 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4384 358 Recordset
4385 359 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4386 360 Recordset
4387 361 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4388 362 Recordset
4389 363 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4390 364 Recordset
4391 365 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4392 366 Recordset
4393 367 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4394 368 Recordset
4395 369 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4396 370 Recordset
4397 371 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4398 372 Recordset
4399 373 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4400 374 Recordset
4401 375 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4402 376 Recordset
4403 377 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4404 378 Recordset
4405 379 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4406 380 Recordset
4407 381 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4408 382 Recordset
4409 383 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4410 384 Recordset
4411 385 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4412 386 Recordset
4413 387 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4414 388 Recordset
4415 389 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4416 390 Recordset
4417 391 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4418 392 Recordset
4419 393 Dim recMandProp As Recordset, recPEF As Recordset, recMandProp As
4420 394 Recordset

```

```

4420 380      recMarClsFit.MoveFirst
4421 381      End If
4422 382      * Check to see if next record is the same school as the last
4423 383      * If it is, append it to the MAR_CLS_FIT table
4424 384      If (recClass.SchoolNumber_FK = strClassNumber) And
4425 385      Application.SeOption("Confirm Action Queries") = False
4426 386      db1.Execute "INSERT INTO MAR_CLS_FIT (SSN_FK,
4427 387      MarineIndex, ClassIndex_FK, Fitness) SELECT TEMP_FIT.SSN_FK,
4428 388      TEMP_FIT.MarineIndex, TEMP_FIT.ClassIndex_FK, TEMP_FIT.Fitness FROM
4429 389      TEMP_FIT WHERE TEMP_FIT.Fitness > 0;"
4430 390      Application.SeOption("Confirm Action Queries") = True
4431 391      recClass.MoveNext
4432 392      Else
4433 393      angSumOfFit = 0
4434 394      angCount = 0
4435 395      boolPEF = False
4436 396      Verify recMarine has values
4437 397      If recMarine.RecordCount > 0 Then
4438 398      * Initialize variables
4439 399      strInput1 = "" & recClass.SchoolNumber_FK & ""
4440 400      strInput2 = "" & recClass.AMOS_FK & ""
4441 401      intInput = 0
4442 402      * Find all PEFs associated with the current school
4443 403      strSQL = "SELECT PEF_FK FROM SCH_PROP WHERE
4444 404      SchoolNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4445 405      Set recPEF = db1.OpenRecordset(strSQL)
4446 406      If recPEF.EOF = False Then
4447 407      recPEF.MoveLast
4448 408      recPEF.MoveFirst
4449 409      End If
4450 410      * Find mandatory fundamental properties associated
4451 411      with the current school
4452 412      strSQL = "SELECT FProperty.Name_FK FROM FUND_SCH_PROP
4453 413      WHERE SchoolNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4454 414      AND Level = " & intInput & ""
4455 415      Set recFProp = db1.OpenRecordset(strSQL)
4456 416      If recFProp.EOF = False Then
4457 417      recFProp.MoveLast
4458 418      recFProp.MoveFirst
4459 419      End If
4460 420      * Find mandatory logical properties associated with
4461 421      the current school
4462 422      strSQL = "SELECT LProperty.Name_FK FROM LOG_SCH_PROP
4463 423      WHERE SchoolNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4464 424      AND Level = " & intInput & ""
4465 425      Set recLProp = db1.OpenRecordset(strSQL)
4466 426      If recLProp.EOF = False Then
4467 427      recLProp.MoveLast
4468 428      recLProp.MoveFirst
4469 429      End If
4470 430      * Find desired fundamental properties associated with
4471 431      the current school
4472 432      strSQL = "SELECT FProperty.Name_FK, [Level] FROM
4473 433      FUND_SCH_PROP WHERE SchoolNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4474 434      AND Level = " & intInput & ""
4475 435      Set recFDesired = db1.OpenRecordset(strSQL)
4476 436      If recFDesired.EOF = False Then
4477 437      recFDesired.MoveLast
4478 438      recFDesired.MoveFirst
4479 439      End If
4480 440      * Find desired logical properties associated with the
4481 441      current school
4482 442      strSQL = "SELECT LProperty.Name_FK, [Level] FROM
4483 443      LOG_SCH_PROP WHERE SchoolNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4484 444      AND Level = " & intInput & ""
4485 445      Set recLDesired = db1.OpenRecordset(strSQL)
4486 446      If recLDesired.EOF = False Then
4487 447      recLDesired.MoveLast
4488 448      recLDesired.MoveFirst
4489 449      End If
4490 450      * Cycle through all the marines
4491 451      For j = 1 To recMarine.RecordCount
4492 452      * Yield control temporarily to Windows
4493 453      DoEvents
4494 454      * Check to see if marine's PEF is an open
4495 455      If recMarine.PEF < "00" Then
4496 456      boolPEF = False
4497 457      If recPEF.EOF = False Then
4498 458      * Check to see if marine's PEF is
4499 459      associated with the school
4500 460      For k = 1 To recPEF.RecordCount
4501 461      If recPEF.PEF_FK = recMarine.PEF Then
4502 462      boolPEF = True
4503 463      End If
4504 464      recPEF.MoveNext
4505 465      Next k
4506 466      recPEF.MoveFirst
4507 467      End If
4508 468      Else
4509 469      boolPEF = True
4510 470      End If
4511 471      * Compare marine attributes with the mandatory
4512 472      properties
4513 473      If boolPEF = True Then
4514 474      * Check for mandatory fundamental properties
4515 475      If recFDesired.EOF = False Then
4516 476      For k = 1 To recFDesired.RecordCount
4517 477      * All tests must prove true for the
4518 478      mandatory properties
4519 479      If boolMandProp = True Then
4520 480      FundPropTest(recFDesired.FProperty.Name_FK, recMarine.SSN_PK) = False Then
4521 481      boolMandProp = False
4522 482      End If
4523 483      recFDesired.MoveNext
4524 484      End If
4525 485      Next k
4526 486      recFDesired.MoveFirst
4527 487      End If
4528 488      * Check for mandatory logical properties
4529 489      If boolMandProp = True Then
4530 490      If recLDesired.EOF = False Then
4531 491      For k = 1 To recLDesired.RecordCount
4532 492      * All tests must prove true for the
4533 493      mandatory properties
4534 494      If boolMandProp = True Then
4535 495      LogPropTest(recLDesired.LProperty.Name_FK, recMarine.SSN_PK) = False Then
4536 496      boolMandProp = False
4537 497      End If
4538 498      recLDesired.MoveNext
4539 499      End If
4540 500      End If
4541 501      End If
4542 502      Next j
4543 503      End If
4544 504      End If
4545 505      End If
4546 506      End If
4547 507      End If
4548 508      End If
4549 509      End If
4550 510      End If
4551 511      End If
4552 512      End If
4553 513      End If
4554 514      End If
4555 515      End If
4556 516      End If
4557 517      End If
4558 518      End If
4559 519      End If
4560 520      End If
4561 521      End If
4562 522      End If
4563 523      End If
4564 524      End If
4565 525      End If
4566 526      End If
4567 527      End If
4568 528      End If
4569 529      End If
4570 530      End If
4571 531      End If
4572 532      End If
4573 533      End If
4574 534      End If
4575 535      End If
4576 536      End If
4577 537      End If
4578 538      End If
4579 539      End If
4580 540      End If
4581 541      End If
4582 542      End If
4583 543      End If
4584 544      End If
4585 545      End If
4586 546      End If
4587 547      End If
4588 548      End If
4589 549      End If
4590 550      End If
4591 551      End If
4592 552      End If
4593 553      End If
4594 554      End If
4595 555      End If
4596 556      End If
4597 557      End If
4598 558      End If
4599 559      End If
4600 560      End If
4601 561      End If
4602 562      End If
4603 563      End If
4604 564      End If
4605 565      End If
4606 566      End If
4607 567      End If
4608 568      End If
4609 569      End If
4610 570      End If
4611 571      End If
4612 572      End If
4613 573      End If
4614 574      End If
4615 575      End If
4616 576      End If
4617 577      End If
4618 578      End If
4619 579      End If
4620 580      End If
4621 581      End If
4622 582      End If
4623 583      End If
4624 584      End If
4625 585      End If
4626 586      End If
4627 587      End If
4628 588      End If
4629 589      End If
4630 590      End If
4631 591      End If
4632 592      End If
4633 593      End If
4634 594      End If
4635 595      End If
4636 596      End If
4637 597      End If
4638 598      End If
4639 599      End If
4640 600      End If
4641 601      End If
4642 602      End If
4643 603      End If
4644 604      End If
4645 605      End If
4646 606      End If
4647 607      End If
4648 608      End If
4649 609      End If
4650 610      End If
4651 611      End If
4652 612      End If
4653 613      End If
4654 614      End If
4655 615      End If
4656 616      End If
4657 617      End If
4658 618      End If
4659 619      End If
4660 620      End If
4661 621      End If
4662 622      End If
4663 623      End If
4664 624      End If
4665 625      End If
4666 626      End If
4667 627      End If
4668 628      End If
4669 629      End If
4670 630      End If
4671 631      End If
4672 632      End If
4673 633      End If
4674 634      End If
4675 635      End If
4676 636      End If
4677 637      End If
4678 638      End If
4679 639      End If
4680 640      End If
4681 641      End If
4682 642      End If
4683 643      End If
4684 644      End If
4685 645      End If
4686 646      End If
4687 647      End If
4688 648      End If
4689 649      End If
4690 650      End If
4691 651      End If
4692 652      End If
4693 653      End If
4694 654      End If
4695 655      End If
4696 656      End If
4697 657      End If
4698 658      End If
4699 659      End If
4700 660      End If
4701 661      End If
4702 662      End If
4703 663      End If
4704 664      End If
4705 665      End If
4706 666      End If
4707 667      End If
4708 668      End If
4709 669      End If
4710 670      End If
4711 671      End If
4712 672      End If
4713 673      End If
4714 674      End If
4715 675      End If
4716 676      End If
4717 677      End If
4718 678      End If
4719 679      End If
4720 680      End If
4721 681      End If
4722 682      End If
4723 683      End If
4724 684      End If
4725 685      End If
4726 686      End If
4727 687      End If
4728 688      End If
4729 689      End If
4730 690      End If
4731 691      End If
4732 692      End If
4733 693      End If
4734 694      End If
4735 695      End If
4736 696      End If
4737 697      End If
4738 698      End If
4739 699      End If
4740 700      End If
4741 701      End If
4742 702      End If
4743 703      End If
4744 704      End If
4745 705      End If
4746 706      End If
4747 707      End If
4748 708      End If
4749 709      End If
4750 710      End If
4751 711      End If
4752 712      End If
4753 713      End If
4754 714      End If
4755 715      End If
4756 716      End If
4757 717      End If
4758 718      End If
4759 719      End If
4760 720      End If
4761 721      End If
4762 722      End If
4763 723      End If
4764 724      End If
4765 725      End If
4766 726      End If
4767 727      End If
4768 728      End If
4769 729      End If
4770 730      End If
4771 731      End If
4772 732      End If
4773 733      End If
4774 734      End If
4775 735      End If
4776 736      End If
4777 737      End If
4778 738      End If
4779 739      End If
4780 740      End If
4781 741      End If
4782 742      End If
4783 743      End If
4784 744      End If
4785 745      End If
4786 746      End If
4787 747      End If
4788 748      End If
4789 749      End If
4790 750      End If
4791 751      End If
4792 752      End If
4793 753      End If
4794 754      End If
4795 755      End If
4796 756      End If
4797 757      End If
4798 758      End If
4799 759      End If
4800 760      End If
4801 761      End If
4802 762      End If
4803 763      End If
4804 764      End If
4805 765      End If
4806 766      End If
4807 767      End If
4808 768      End If
4809 769      End If
4810 770      End If
4811 771      End If
4812 772      End If
4813 773      End If
4814 774      End If
4815 775      End If
4816 776      End If
4817 777      End If
4818 778      End If
4819 779      End If
4820 780      End If
4821 781      End If
4822 782      End If
4823 783      End If
4824 784      End If
4825 785      End If
4826 786      End If
4827 787      End If
4828 788      End If
4829 789      End If
4830 790      End If
4831 791      End If
4832 792      End If
4833 793      End If
4834 794      End If
4835 795      End If
4836 796      End If
4837 797      End If
4838 798      End If
4839 799      End If
4840 800      End If
4841 801      End If
4842 802      End If
4843 803      End If
4844 804      End If
4845 805      End If
4846 806      End If
4847 807      End If
4848 808      End If
4849 809      End If
4850 810      End If
4851 811      End If
4852 812      End If
4853 813      End If
4854 814      End If
4855 815      End If
4856 816      End If
4857 817      End If
4858 818      End If
4859 819      End If
4860 820      End If
4861 821      End If
4862 822      End If
4863 823      End If
4864 824      End If
4865 825      End If
4866 826      End If
4867 827      End If
4868 828      End If
4869 829      End If
4870 830      End If
4871 831      End If
4872 832      End If
4873 833      End If
4874 834      End If
4875 835      End If
4876 836      End If
4877 837      End If
4878 838      End If
4879 839      End If
4880 840      End If
4881 841      End If
4882 842      End If
4883 843      End If
4884 844      End If
4885 845      End If
4886 846      End If
4887 847      End If
4888 848      End If
4889 849      End If
4890 850      End If
4891 851      End If
4892 852      End If
4893 853      End If
4894 854      End If
4895 855      End If
4896 856      End If
4897 857      End If
4898 858      End If
4899 859      End If
4900 860      End If
4901 861      End If
4902 862      End If
4903 863      End If
4904 864      End If
4905 865      End If
4906 866      End If
4907 867      End If
4908 868      End If
4909 869      End If
4910 870      End If
4911 871      End If
4912 872      End If
4913 873      End If
4914 874      End If
4915 875      End If
4916 876      End If
4917 877      End If
4918 878      End If
4919 879      End If
4920 880      End If
4921 881      End If
4922 882      End If
4923 883      End If
4924 884      End If
4925 885      End If
4926 886      End If
4927 887      End If
4928 888      End If
4929 889      End If
4930 890      End If
4931 891      End If
4932 892      End If
4933 893      End If
4934 894      End If
4935 895      End If
4936 896      End If
4937 897      End If
4938 898      End If
4939 899      End If
4940 900      End If
4941 901      End If
4942 902      End If
4943 903      End If
4944 904      End If
4945 905      End If
4946 906      End If
4947 907      End If
4948 908      End If
4949 909      End If
4950 910      End If
4951 911      End If
4952 912      End If
4953 913      End If
4954 914      End If
4955 915      End If
4956 916      End If
4957 917      End If
4958 918      End If
4959 919      End If
4960 920      End If
4961 921      End If
4962 922      End If
4963 923      End If
4964 924      End If
4965 925      End If
4966 926      End If
4967 927      End If
4968 928      End If
4969 929      End If
4970 930      End If
4971 931      End If
4972 932      End If
4973 933      End If
4974 934      End If
4975 935      End If
4976 936      End If
4977 937      End If
4978 938      End If
4979 939      End If
4980 940      End If
4981 941      End If
4982 942      End If
4983 943      End If
4984 944      End If
4985 945      End If
4986 946      End If
4987 947      End If
4988 948      End If
4989 949      End If
4990 950      End If
4991 951      End If
4992 952      End If
4993 953      End If
4994 954      End If
4995 955      End If
4996 956      End If
4997 957      End If
4998 958      End If
4999 959      End If

```

```

4680 596 recMarClsRt.Close
4681 597
4682 598
4683 599 ' Prepare the AMPL_rdm1.dtl data file
4684 600 Application.SetOption "Confirm Action Queries", False
4685 601 db1.Execute "DELETE " FROM TEMP_FIT;"
4686 602 ' Specify the number of marlines
4687 603 db1.Execute "INSERT INTO TEMP_FIT (SSN_PK, MarlineIndex) VALUES
4688 604 (param totalMarlines - 1, & recMarlineRecordCount & ");"
4689 605 ' Specify the number of school classes
4690 606 db1.Execute "INSERT INTO TEMP_FIT (SSN_PK, MarlineIndex) VALUES
4691 607 (param totalClasses - 1, & recClassRecordCount & ");"
4692 608 ' Specify the class demand
4693 609 db1.Execute "INSERT INTO TEMP_FIT (SSN_PK) VALUES (param demand
4694 610 db1.Execute "INSERT INTO TEMP_FIT (ClassIndex, PK_Fitness) SELECT
4695 611 CLASS.ClassIndex, CLASS.Quota FROM CLASS;"
4696 612 ' Specify the class penalty
4697 613 db1.Execute "INSERT INTO TEMP_FIT (SSN_PK) VALUES (param penalty
4698 614 db1.Execute "INSERT INTO TEMP_FIT (ClassIndex, PK_Fitness) SELECT
4699 615 PENALTY.ClassIndex, PK_PENALTY.Penalty FROM PENALTY;"
4700 616 DoCmd.TransferText acExportFixed, "rdm1_Data Export Specification",
4701 617 TEMP_FIT, "C:\RDMAmpData\rdm1.dtl"
4702 618
4703 619 ' Prepare the AMPL_rdm2.dtl data file
4704 620 db1.Execute "DELETE " FROM TEMP_FIT;"
4705 621 db1.Execute "INSERT INTO TEMP_FIT SELECT * FROM MAR_CLS_FIT;"
4706 622 db1.Execute "ALTER TABLE TEMP_FIT DROP COLUMN SSN_PK;"
4707 623 db1.Execute "ALTER TABLE TEMP_FIT ADD COLUMN SSN_PK text;"
4708 624 db1.Execute "INSERT INTO TEMP_FIT (SSN_PK) VALUES (param fitness
4709 625 default 0 & ");"
4710 626 db1.Execute "SELECT * INTO AMPL_TEMP FROM TEMP_FIT ORDER BY SSN_PK
4711 627 db1.Execute "INSERT INTO AMPL_TEMP (SSN_PK) VALUES (");"
4712 628 DoCmd.TransferText acExportFixed, "rdm2_Data Export Specification",
4713 629 AMPL_TEMP, "C:\RDMAmpData\rdm2.dtl"
4714 630 Application.SetOption "Confirm Action Queries", True
4715 631
4716 632 recMarline.Close
4717 633 recClass.Close
4718 634
4719 635 ' Remove the created tables
4720 636 db1.Execute "DROP TABLE TEMP_FIT;"
4721 637 db1.Execute "DROP TABLE AMPL_TEMP;"
4722 638 db1.Execute "Drop Table TEMP_FIT;"
4723 639 db1.Close
4724 640 ' Remove the meter from the status bar
4725 641 varRetVal = SysCmd(acSysCmdRemoveMeter)
4726 642
4727 643 End Sub
4728 644
4729 645
4730 646
4731 647
4732 648
4733 649 Public Sub QuoPenalty(varDateBound As Variant)
4734 650 Dim varDate
4735 651 Dim rec As Recordset, rec1 As Recordset
4736 652 Dim strSQL As String, strConv As String
4737 653 Dim i As Integer
4738 654 Dim db1 As Database
4739 655
4740 656 ' Create new class quota file
4741 657 Application.SetOption "Confirm Action Queries", False
4742 658
4743 659 ' Delete the old penalty and class files
4744 660 DoCmd.RunSQL "DELETE " FROM PENALTY;"
4745 661 DoCmd.RunSQL "DELETE " FROM CLASS;"
4746 662
4747 663 ' Query generates new class quota
4748 664 DoCmd.SetWarnings (False)
4749 665 DoCmd.OpenQuery "qryNewQuota"
4750 666 DoCmd.SetWarnings (True)
4751 667
4752 668 ' Delete any values < 0
4753 669 DoCmd.RunSQL "DELETE * FROM CLASS WHERE Quota <= 0;"
4754 670
4755 671 Application.SetOption "Confirm Action Queries", True
4756 672
4757 673 ' Create index for the class table and format the FY for RD3 file
4758 674 Set db1 = CurrentDb()
4759 675 strSQL = "SELECT CLASS, SCHOOL.PenaltyFactor FROM SCHOOL INNER JOIN
4760 676 CLASS ON (SCHOOL.AMCS_PK = CLASS.AMCS_PK) AND (SCHOOL.SchoolNumber_PK =
4761 677 CLASS.SchoolNumber_PK)"
4762 678 Set rec = db1.OpenRecordset(strSQL, dbOpenDynaset)
4763 679 rec.MoveLast
4764 680
4765 681 If rec.RecordCount > 0 Then
4766 682 rec.MoveFirst
4767 683 For i = 1 To rec.RecordCount
4768 684 rec.Edit
4769 685 rec1 = CurrentDb().OpenRecordset("SELECT * FROM CLASS WHERE Quota <= 0")
4770 686 rec1.MoveFirst
4771 687 ' Converts the fiscal year to a two digit number for RD3 file
4772 688 strConv = CStr(rec1.FiscalYear_PK)
4773 689 strConv = Right(strConv, 2)
4774 690 rec1.FiscalYear_PK = CInt(strConv)
4775 691 rec.Update
4776 692 rec.MoveNext
4777 693 Next i
4778 694 End If
4779 695
4780 696 ' Create penalty file
4781 697 ' First, update the index from the CLASS table
4782 698 Application.SetOption "Confirm Action Queries", False
4783 699 DoCmd.OpenQuery "qryPenaltyIndex"
4784 700 Application.SetOption "Confirm Action Queries", True
4785 701 ' Create a record set of the PENALTY table
4786 702 strSQL = "SELECT * FROM PENALTY;"
4787 703 Set rec1 = db1.OpenRecordset(strSQL, dbOpenDynaset)
4788 704 If Not (varDateBound = False) Then
4789 705 ' Format this date for use in the DateDiff function
4790 706 varDateBound = Format(varDateBound, "####/##/##")
4791 707
4792 708 Module: modFitnessDetermination Page: 16
4793 709
4794 710 If rec.RecordCount > 0 Then
4795 711 rec.MoveFirst
4796 712 For i = 1 To rec.RecordCount
4797 713 rec1 = CurrentDb().OpenRecordset("SELECT * FROM CLASS WHERE Quota <= 0")
4798 714 rec1.MoveFirst
4799 715 ' Get report date from class record
4800 716 varDate = rec1.ReportDate
4801 717 ' Format the report date for use in the DateDiff function
4802 718 varDate = Format(varDate, "####/##/##")
4803 719 ' Make the penalty the difference between the two values
4804 720 ' This makes the penalty linear, based on date
4805 721 rec1.Penalty = DateDiff("d", varDate, varDateBound) *
4806 722 rec1.Update
4807 723 rec.MoveNext
4808 724 Next i
4809 725 End If

```

```

4939 78 DoCmd.RunSQL "DELETE DISTINCTROW MARINE." & ASSIGNMENT.FK FROM MARINE
4940 INNER JOIN ASSIGNMENT ON MARINE.SSN_PK = ASSIGNMENT.SSN_FK WHERE
4941 ASSIGNMENT.ReportDate < " & varTodaysDate & "
4942 79
4943 80 Select Case MsgBox("Delete unassigned Marines from the database?",
4944 vbYesNo + vbQuestion, "Delete Unassigned Marines")
4945 81 Case 0 " If yes is clicked
4946 82 " Deletes entries in the Marine table without a corresponding
4947 entry in the Assignment table
4948 83 DoCmd.RunSQL "DELETE DISTINCTROW MARINE." & ASSIGNMENT.SSN_FK

```

```

4949 FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK = ASSIGNMENT.SSN_FK
4950 WHERE (((ASSIGNMENT.SSN_FK) Is Null));"
4951 84 Case 1 " If no is clicked
4952 85 " No action taken
4953 86 End Select
4954 87
4955 88 End Sub

```

LIST OF REFERENCES

- [1] USMC, Manpower Management Information Systems Branch, Manpower and Reserve Affairs Department. "Statement of Work for the USMC Manpower Model Modernization: Reengineering of the Enlisted Assignment Model and Recruit Distribution Model," September, 1997.
- [2] Decision Support Associates Inc. *Users Manual-Recruit Distribution Model*. Developed In-house, Decision Support Associates Inc., 1993.
- [3] C.J. Date. *An Introduction to Database Systems*. Reading, MA: Addison-Wesley, 1992.
- [4] J. Dumas. *Designing User Interfaces for Software*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [5] CPLEX, A Division of ILOG. CPLEX Home Page, 1998. [<http://www.cplex.com>].
- [6] R.H. Sprague. "A Framework for the Development of Decision Support Systems." *MIS Quarterly* 4, Pages 1-26, 1980.
- [7] Logic Works, Inc. "Logic Works BPWin Tutorial Guide," 1994-1995.
- [8] E.F. Codd. "A Relational Model of Data for Large Shared Databanks." *Communications of the ACM* 13, June 1970.
- [9] R. Smith, D. Sussman. *Beginning Access 97 VBA Programming*. Wrox Press, 1997.
- [10] R. Fourer, D. Gay, B. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Boyd & Fraser Publishing Company, 1993.
- [11] B. Render, R. Stair. *Quantitative Analysis for Management*. Prentice-Hall Inc., 1997.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
8725 John J. Kingman Rd., Ste 0944
Fort Belvoir, VA 22060-6218
2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943-5101
3. Professor Hemant Bhargava (Code SM/BH) 7
Systems Management Department
Naval Postgraduate School
Monterey, CA 93940-5000
4. Professor Suresh Sridhar (Code SM/SR) 1
Systems Management Department
Naval Postgraduate School
Monterey, CA 93943-5000
5. Professor Daniel Dolk (Code SM/Dk) 1
Systems Management Department
Naval Postgraduate School
Monterey, CA 93943-5000
6. LT Kevin J. Snoap 2
541 Westway Drive N.W.
Grand Rapids, MI 49544